

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

semanal

AÑO II N.º 31

160 Ptas.

Canarias 165 pts.

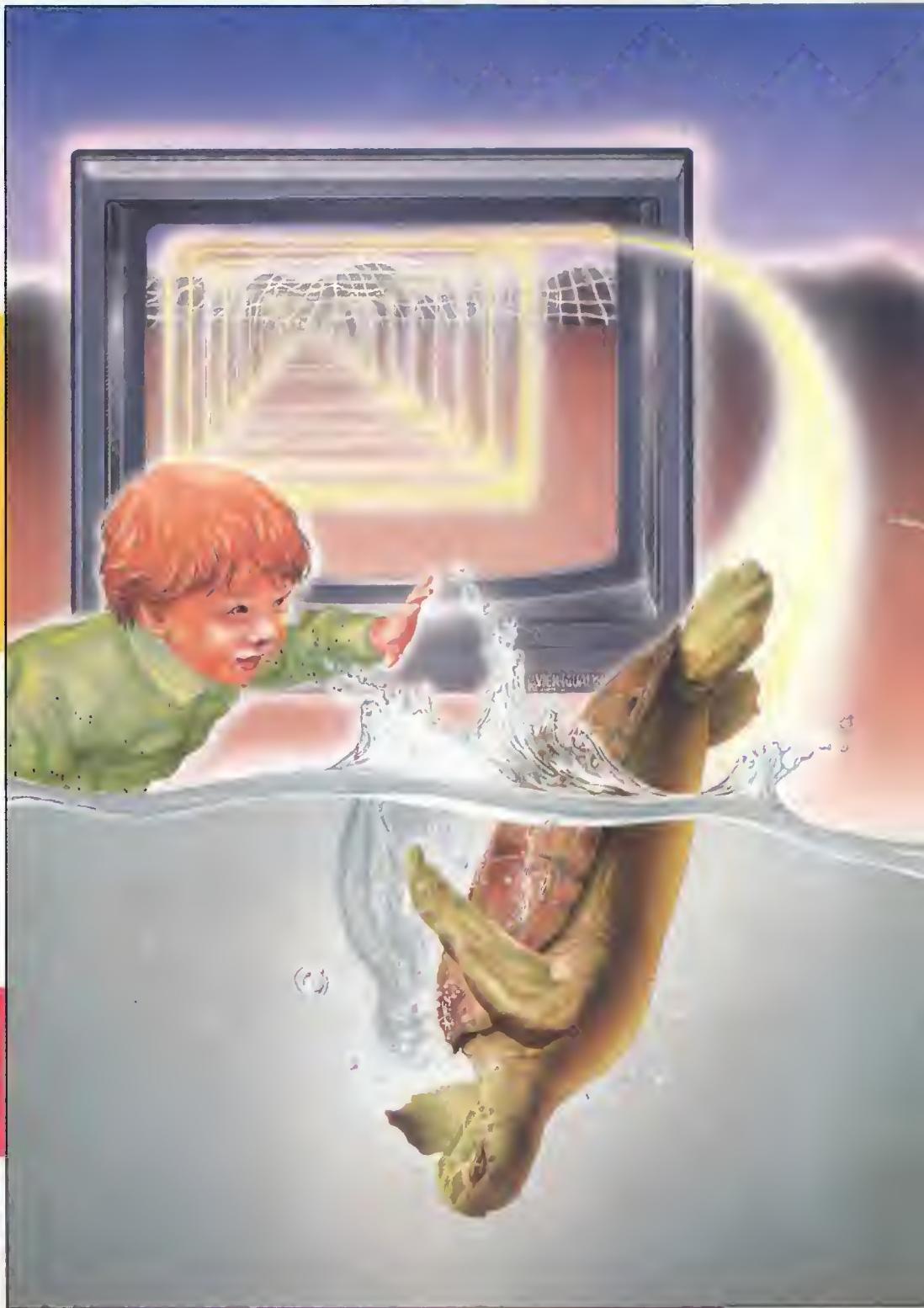
PROBAMOS
EN EXCLUSIVA
LAS AMPLIACIONES
DE MEMORIA PARA
LOS CPC

**LENGUAJE
LOGO:
APRENDER
A PENSAR...
JUGANDO**

**¡SENSACIONAL
CONCURSO!
CONSIGUE GRATIS,
DURANTE UN AÑO,
LOS MEJORES JUEGOS
DEL MERCADO
PARA TU AMSTRAD**

**CONTROL DE STOCKS
PARA EL PCW8256,
EN MALLARD BASIC
Y... GRATIS**

**AMSTRAD
YA TIENE
TARJETAS EPROM
DE UN MEGABYTE**



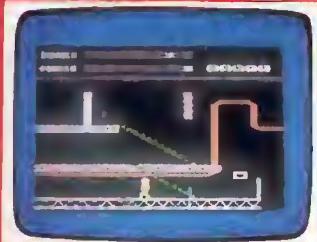


MICROBYTE PRESENTA...

NUEVOS PROGRAMAS EN CASSETTE Y DISCO

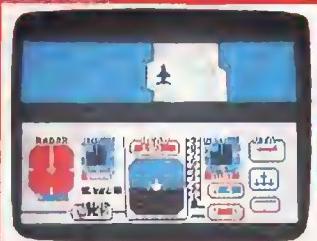
AMSTRAD

ARGO NAVIS



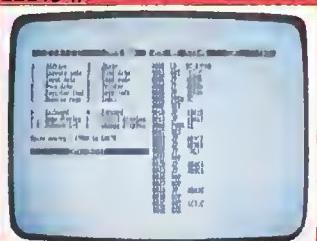
El comandante de nave AMSTRAD-1 se encuentra atrapado en las profundidades de una central nuclear y debe salir con vida. Excelentes gráficos y sonido. P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

JUMP JET



Te encuentras a los mandos de la nave "Aircraft". En una perfecta maniobra debes despegar del portaviones. (Excelente versión simulador vuelo-combate). P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

ZEDIS II



Editor-desensamblador del Z-80, para el programador más avanzado. P.V.P.: CASSETTE 1.900 pts. DISCO 2.600 pts.

OTROS PROGRAMAS EN STOCK

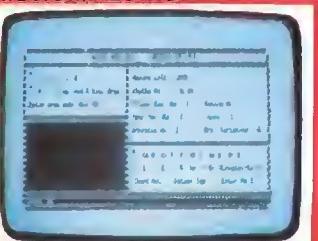
MINI OFFICE	P.V.P. CASS. 3.200 pts. P.V.P. DIS. 3.900 pts.
WORLD CUP FOOTBALL	P.V.P. CASS. 1.800 pts.
BATTLE FOR MIDWAY	P.V.P. CASS. 1.800 pts.
FIGHTER PILOT	P.V.P. CASS. 2.200 pts.
SURVIVOR	P.V.P. CASS. 1.800 pts.
MOON BUGGY	P.V.P. CASS. 1.800 pts.
TECHNICIAN TED	P.V.P. CASS. 1.800 pts.
FRUITY FRANK	P.V.P. CASS. 1.800 pts.
DATABASE	P.V.P. CASS. 2.100 pts.
LOGO TURTLE GRAPHICS	P.V.P. CASS. 2.400 pts.
TASCOPY Y TASPRINT	P.V.P. CASS. 2.600 pts.
FONT EDITOR	P.V.P. CASS. 1.900 pts.

ROCK RAID



Debes pilotar con acierto la nave que a lo largo de su viaje galáctico sufrirá encuentros con meteoritos, residuos planetarios, etc. Gran movilidad y excelentes efectos. P.V.P.: CASSETTE 1.900 pts. DISCO 2.600 pts.

MUSIC MAESTRO



El más completo programa de música creado para el AMSTRAD. Permite crear sonidos, melodías y convertir tu ordenador en la mejor "caja de música". P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

PAZAZZ

WIZARD'S LAIR



Te encuentras atrapado en las profundidades de una caverna, llena de obstáculos, adversidades, etc. ¿Serás capaz de salir con vida? P.V.P.: CASSETTE 1.900 pts. DISCO 2.600 pts.

PAZAZZ



Programa que permite de una manera sencilla la creación de pantallas con gráficos, dotados de movimiento, acompañados de música. P.V.P.: DISCO 2.900 pts.

MACADAM FLIPPER



Avanzado programa que nos traslada al manejo de la máquina-flipper del mejor casino de Las Vegas. Posibilidad de creación del tablero, puntuaciones, etc. P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

SYCLONE 2



Programa de utilidad que permite realizar copias de seguridad (back-up) a distintas velocidades (baudios). P.V.P.: CASSETTE 800 pts. DISCO 2.500 pts.

SYSTEM X



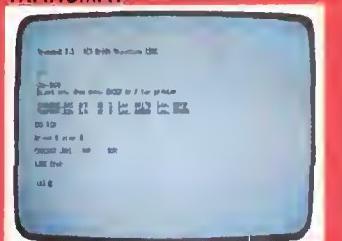
Ampliación del lenguaje BASIC. Conjunto de 30 nuevas instrucciones (fill, circle, errectc) para ayudar en la programación. P.V.P.: CASSETTE 2.200 pts. DISCO 2.900 pts.

ODDJOB



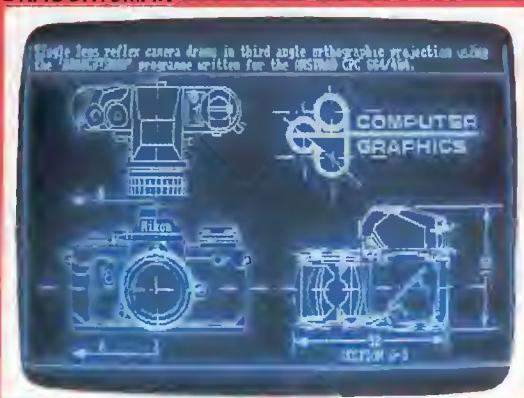
La mayor utilidad para el mejor conocimiento del disco. (Copias de disco, Disk map, Disk track, sector, etc.). P.V.P.: DISCO 2.600 pts.

TRANSMAT



Pasar los mejores programas de cinta a disco ya no es problema. Con Transmat este proceso será fácil y sencillo. P.V.P.: DISCO 2.600 pts.

DRAUGHTSMAN



Sistema de plóter para dibujar en tercera dimensión. Programa escrito para el AMSTRAD CPC 464/664.

ENVINOS A MICROBYTE AS.

P.º Castellana, 179, 1.º - 28046 Madrid

Nombre _____

Apellidos _____

Dirección _____

Población _____

D.P. _____

Teléfono _____

ENVIOS GRATIS

JUEGO	C	D	Preco	TOTAL

PRECIO TOTAL PESETAS

Incluye teléfono nominativo _____

Contra-Reembolso _____

Pedidos por teléfono 91 - 442 54 33 / 44

MICROHOBBY

AMSTRAD

SUPERARIO

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

José M. Díaz

Redactor Jefe

Juan José Martínez

Diseño gráfico

José Flores

Colaboradores

Javier Borceló

David Sopuerta

Robert Chotwin

Eduardo Ruiz

Francois Partalo

Pedro Sudón

Miguel Sepúlvedo

Francois Martín

Jesús Alonso

Pedro S. Pérez

Amilio Gómez

Secretaria Redacción

Carmen Sontomorio

Fotografía

Carlos Condel

Portada

Javier Igual

Ilustradores

J. Igual, J. Pons, F. L. Frantón,
J. Septién, Peja, J. J. Mora

Edita

HOBBY PRESS, S.A.

Presidente

Maria Andriño

Consejero Delegado

José I. Gómez-Centurión

Jefe de Producción

Carlos Peropodre

Marketing

Marta García

Jefe de Publicidad

Concho Gutiérrez

Publicidad Barcelona

José Golán Cortés

Tel: (93) 303 10 22/313 71 62

Secretario de Dirección

Mario Cogoró

Suscripciones

M.º Rosa González

M.º del Mar Calzado

Redacción, Administración y Publicidad

Lo Gronjo, 39

Polygono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Poulinio Blanco

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDIC, S. A. Crta. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solana, 16

Depósito Legal:

M-28468-1985

Derechos exclusivos
de la revista

**COMPUTING with
the AMSTRAD**

Representante para Argentina, Chile,
Uruguay y Paraguay, Cia.

Americana de Ediciones, S.R.L. Sud

América 1.532, Tel.: 21 24 64. 1209

BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace
necesariamente solidaria de los
opiniones vertidas por sus
colaboradores en los artículos
firmados. Reservados todos los
derechos.

Se solicitará control OJD

Año II • Número 31 1 al 7 de Abril de 1986
160 ptas. (incluida I.V.A.)
Canarias, 155 ptas. + 10 ptas. sabretasa aérea
Ceuta y Melilla, 155 ptas.

4 Primera plana

Tarjetas EPROM para Amstrad.

Logo 6

Tadas las que tenemos un Amstrad con disco, tenemos también un lenguaje Logo de Digital Research. Ya es hora de aprender a manejarlo y, sobre todo, comprender por qué hay que hacerla.



10 Primeros pasos

En este artículo se resume, mediante una aplicación práctica, toda la que hemos aprendido, y más, acerca de los gráficos del Amstrad.

14 Análisis



18 Mr. Joystick

Ocean nos viene con una nueva y apasionante aventura de la familia de Wally, esta vez en las luxuriantes y oscuras profundidades de la selva tropical.



20 Banco de pruebas

Analizamos exhaustivamente las ampliaciones de memoria de DK'Tronics para los Amstrad de la serie CPC.

26 Para... PCW

Os ofrecemos un completa programa de control de stocks para el PCW escrito completamente en Mallard Basic. El 8256 puede hacer muchas más cosas aparte de procesar textas, precisamente así...

¡No estamos para juegos!

LO NUESTRO ES HACER BUENAS GESTIONES

Para AMSTRAD 8256 y 6128

CONTROL DE ALMACEN + I.V.A.

Sepa lo que tiene, su costo, proveedor, totales por artículos y general. Pida cuantos resultados quiera de su explotación, por producto, proveedor, etc., etc. (15.300 ptas. incluido I.V.A.)

CUENTAS PROVEEDORES-BANCOS-CLIENTES

Sencillo control de cuentas separadas (cuantas quiera), con lo que eliminará el problema diario de la pequeña empresa con buen movimiento (8.600 ptas. incluido I.V.A.)

RECIBOS

Programa que resuelve el mecanografiado interminable. Asociaciones, comunidades, colegios, clubes, podrán hacer los recibos normalizados con domiciliación bancaria y posibilidad de correcciones. (8.300 ptas. incluido I.V.A.)

CLIENTES

Datos actualizados, etiquetas correspondencia. Petición por 5 campos los resúmenes (18.600 ptas. incluido I.V.A.)

FACTURACION

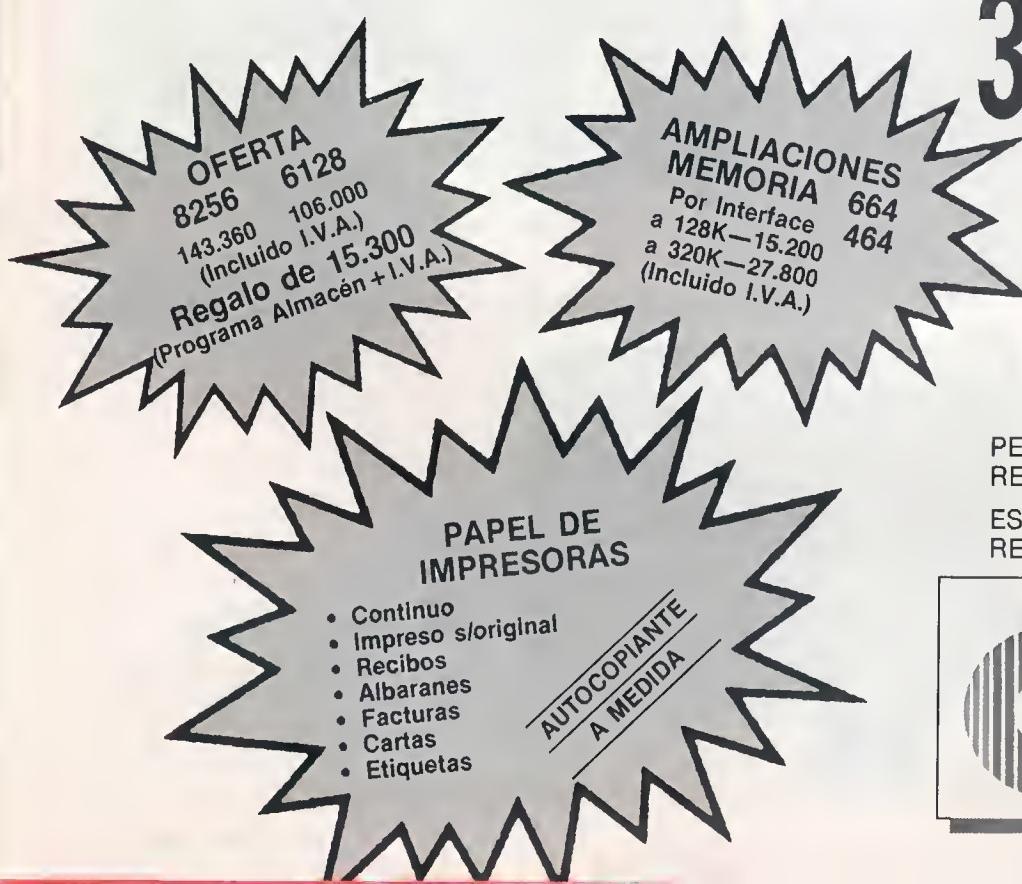
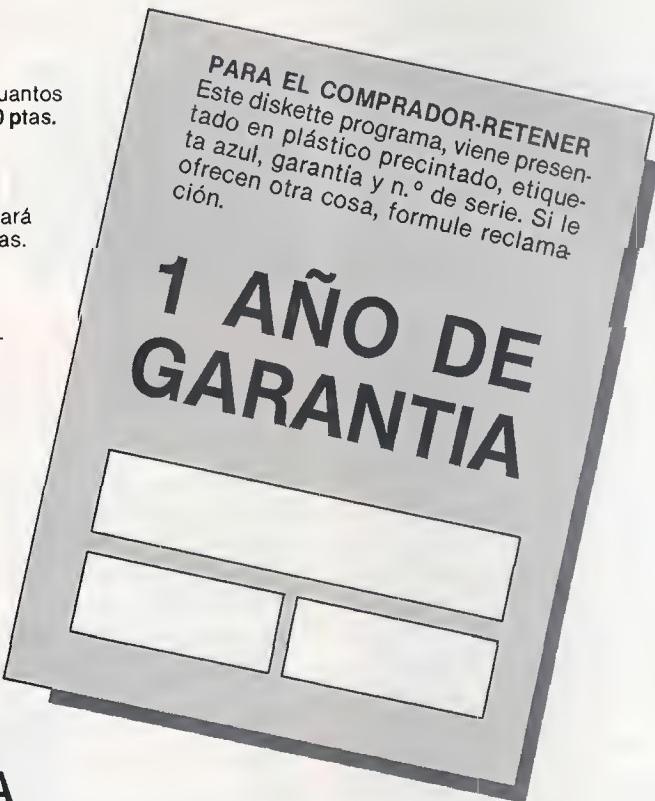
Numeración correlativa automática, fecha automática, resúmenes clientes y totalizaciones (15.300 ptas. incluido I.V.A.)

PRESUPUESTOS

Presupuesto sencillamente cuanto quiera y transfórmelo en factura de forma sencilla. Posibilidad transformaciones y reformas. (18.300 ptas. incluido I.V.A.)

HACEMOS PROGRAMAS A MEDIDA

Encargos llamar o contactar con Juan Luis Ruiz. Tno.: 474 55 32



3 IMPACTOS 3 NOVEDADES 3

PEDIDOS, TELEFONO, CARTA O TELEX
REEMBOLSO SIN GASTOS.

ESPECIAL A COLABORADORES
RESTO DE ESPAÑA



LECTOR DE TARJETAS PARA AMSTRAD

Hace no mucho tiempo, una noticia en el mundo de los ordenadores MSX levantó un gran revuelo: se empezaba a comercializar software en tarjetas EPROM, que se conectaban al bus de expansión del ordenador.

La idea es francamente buena; pensemos, por ejemplo, en los interminables tiempos de carga de los juegos en cassette. Con las tarjetas, podemos empezar a jugar instantáneamente. Basta tan sólo con conectar el ordenador.

Pues bien, esas tarjetas EPROM muy pronto estarán disponibles para **Amstrad**, con unas capacidades de memoria que oscilan entre 16 Kbytes de programa, juegos o aplicaciones, que se cargarán instantáneamente en el **Amstrad** nada más encenderlo.

Las 5 compañías de software más importantes de Inglaterra han firmado un acuerdo para que sus productos aparezcan en este formato.

En España, Serma será la encargada de distribuir este nuevo software, que estará en la calle en abril, a un precio, al menos los juegos, que oscilará entre 4.000 y 5.000 pesetas.

Obviamente, estos «periféricos» serán compatibles con el CPC464, 664 y 6128.

Un detalle más para los «hackers»: uno mismo puede programarse las tarjetas si dispone de un grabador de EPROM.

Serma está en: C/ Bravo Murillo, 377, tercero A. Tel. (91) 733 73 11. Madrid.



Primer PLANA



MINIOFFICE II

Database Publications, además de editar excelentes revistas como «**Computing with the Amstrad**», también hace incursiones en el campo del software.

Concretamente, acaba de ser lanzado en Inglaterra un paquete de software, que podríamos llamar integrado con algo de optimismo, llamado «**Minioffice II**».

Esta serie de programas vienen a mejorar al «**Minioffice I**», que tal vez pasó algo desapercibido, al menos en España.

Minioffice II consta de 6 programas, compatibles con toda la serie CPC:

- Procesador de textos: puede trabajar con 20, 40 y 80 columnas; posee las facultades típicas de estos programas, e incluye además, la posibilidad de mailmerge.
- Base de datos.
- Hoja de cálculo.
- Paquete de gráficos.
- Paquete de comunicaciones.
- Mailing.

Minioffice se vende en Inglaterra a 20 libras, unas 4.000 ptas. aproximadamente, y, según nuestro correspondiente en Londres, el paquete merece la pena. Es bueno.

A propósito, también está disponible en disco, no sólo en cinta.



NOVEDADES DE ABC

ABC. Analog ha relanzado un juego para **Amstrad**, titulado **Chopper Squad**; la historia se desarrolla en el año 1994, y plantea la situación de un indefenso planeta atravesado por las hordas implacables de feroces alienígenas, a las que tú, sólo tú, puedes devolver al Cosmos de una patada en salva sea la parte.

ABC está en:
Santa Cruz de Marcenado, 31. Tel. 248 82 13.

NUESTRA CINTA YA ESTÁ LISTA

La cinta número 6 de **Amstrad Semanal**, conteniendo los programas de los números 17 al 20, está disponible para ser enviada a los lectores que nos la soliciten.



APRENDER A PENSAR... JUGANDO

Logo es una palabra mágica en el mundo de la informática, y responde a las siglas de un lenguaje de programación muy especial, creado ex profeso para educar la mente de las personas, en el sentido de lógica, capacidad de abstracción y creatividad.

Pero Logo, además de ser algo ideal para el aprendizaje de un niño asistido por ordenador, es también un lenguaje de programación de armas tomar, potente y muy sencillo de usar. Por favor, recuérdese siempre esto: cualquier cosa que usted pueda hacer en Basic, puede hacerlo en Logo, y, segundo, tiene ya el lenguaje al comprar un Amstrad con disco. Vamos, pues, a sacarle algún partido.

Javier Igual





cercarse
al Logo por

primera vez, puede resultar más que sorprendente. Quizás, para el Basic-parlante e incluso para los acostumbrados a tratar con lenguas más cultas como el Pascal o el Fortran, los contactos iniciales puedan resultar casi angustiosos. **¿Pero, dónde está el programa?**, es siempre la primera pregunta.

Para el neófito, sin embargo, la cosa se plantea muy distinta, y rápidamente comprende de qué se trata y cómo se programa en Logo. En ocasiones, me he arrimado a un niño que sentado delante de un ordenador se afanaba en realizar su programa, y créanme que he sentido vergüenza. En su mente las ideas debían de aparecer a un ritmo desorbitado, y si no tecleaba más deprisa, no era por que dudase sino por su mala mecanografía. Llegó uno a arrepentirse de esta cabeza cuadruplicada que nos hemos creado a fuerza de «gotos», «runes» y números de línea.

En Logo, el programa, si es que puede emplearse este término, se nos aparece claro, conciso y fluido. Procedimiento a procedimiento, nuestro ordenador va aprendiendo hasta que finalmente podemos decirle, por ejemplo, JUGAR. El ya sabe que jugar significa BARAJAR, REPARTIR, JUGAR BAZA.

Naturalmente, antes habremos tenido que detallarle, de forma particular, cada una de estas palabras, así, JUGAR BAZA podría ser perfectamente ELEGIR CARTA, TIRA EL 1, TIRA EL 2, EVALUAR, PUNTUAR, ROBAR, JUGAR BAZA. Probablemente esto le parezca extraño, como puede ser que en la definición de lo que es JUGAR BAZA aparezcan estas mismas palabras. Esto se conoce como recursividad, quizás por el momento no entienda esto, pero no se preocupe todo se andará.

UN POCO DE HISTORIA

El Logo surge en 1976 de la mano de un grupo de profesionales de la informática y la pedagogía. Walle Feurrieg, Dany Bribons y Seymour Paper son quizás los nombres más significativos asociados a esta historia. Paper había comenzado a desarrollar este lenguaje en 1967, en el Instituto Tecnológico de Massachusetts (MIT). En su realización inicial se empleó un intérprete de Lisp, lenguaje muy utilizado en inteligencia artificial y donde todo se reduce al manejo de listas (*List Processor*).

El Logo, como hijo pequeño de este lenguaje, aún no contando con todas sus bendiciones, sí nos aproxima de forma más que eficaz al proceso de listas. Gracias a este medio se están consiguiendo, en nuestros días grandes progresos en el tratamiento de la información,

como puede ser en las Bases de Datos. A parte de esto, tanto el Lisp como el Logo constituyen espléndidas herramientas para el desarrollo de programas en los que se pretenda simular conversación humana. Es ya un clásico, en este aspecto, el programa Eliza desarrollado en Lisp y perfectamente transportable al Logo. La vocación de Eliza es el sicoanálisis.

Sin embargo, el procesamiento de listas objetivo básico de Logo no es algo que pueda llegar a comprenderse desde el principio. Finalmente, la solución se encontraría en la Tortuga, pequeño robot formado por una campana que podía moverse en todas direcciones y que era capaz de marcar un trazo a medida que avanzaba sobre un papel. Diseñada por Grey Walter, las palabras Logo y Tortuga se convertirían en inseparables. Naturalmente se implementaron las órdenes necesarias para poder manejar a placer la tortuga. Con el tiempo la tortuga acabaría encerrada en la pantalla de los monitores siendo representada por un simple triángulito.

ENSEÑANZA Y LOGO

Tras la beatificación del Logo por parte oficial, en su proyecto Atenea, el campo del primer y segundo ciclo de EGB se encuentra algo revuelto. Pedagogos y profesores se encuentran desorientados tras esta vorágine informática. Para tranquilizar los nervios no estaría de más dar algunas puntuaciones:

1. El lenguaje Logo no se va a introducir como medio de aprender informática, sino como alternativa válida al desarrollo del aprendizaje en el niño. Los cubos rojos que debían apilarse sobre cubos verdes, van a ser sustituidos por un teclado, un monitor y un disco. Así mediante la resolución de distintos problemas se le permite al niño, extraer modelos de razonamiento que más tarde aplicará en su vida cotidiana.

2. El ordenador también va a utilizarse como apoyo en materias concretas. El niño va a poder estudiar geografía, historia o matemáticas tutelado por el orden. Para utilizar estos programas no son necesarios conocimientos de Logo.

3. No se pretende que el niño se convierta en un experto en Logo, aunque naturalmente el peso de su enseñanza debiera recaer en una mezcla de pedagogo y profesional de la informática.



LOGO

CARACTERISTICAS FUNDAMENTALES

El lenguaje Logo es un intérprete, estamos de acuerdo en que esto supone un handicap a la hora de lograr la velocidad de los lenguajes compilados, sin embargo, también es cierto que es mucho más simple el uso de un intérprete. La detección de errores y su corrección es prácticamente inmediata.

Su estructuración es total. Constituido a base de procedimientos (*procedural*), su lectura se hace clara y comprensible. Naturalmente los procedimientos pueden ir acompañados de los parámetros correspondientes. De este modo un procedimiento no tiene que actuar sobre una constante sino que puede ser perfectamente una o distintas variables. Olvídate del tedioso Gosub del Basic y conozca esta forma de subrutinas, la estructuración; ahora sí podemos asegurar que es real. Algo muy parecido a la forma en que se emplean las funciones definidas por el usuario en Basic, con el uso de parámetros.

La recursividad es otra de las características importantes del Logo, quizás en un principio le cueste un poco comprender esto: la idea se basa en llamar a un procedimiento desde el mismo. Un ejemplo de recursividad que siempre suele utilizarse es la definición del factorial de un número. Como ya sabrá el factorial de un número Natural n, no es otra cosa que:

$$n! = n * (n-1) * (n-2) * \dots * 1$$

Definido recursivamente tendríamos:

$$1! = 1$$

$$n! = n * (n-1)! \text{ si } n < > 1$$

Intenta apreciar la diferencia entre ambas definiciones y compárelas. Observa que en la segunda definición el término factorial, término que intentamos definir, también es utilizado.

AMSTRAD Y LOGO

La versión de Logo realizada para **Amstrad**, ha sido desarrollada por Digital Research y corre bajo CPM en sus dos versiones 2.2 y plus. Los poseedores del 664 dispondrán sólo de la primera versión pues la versión 3.0 necesita 128 k.

Para cargar DR. Logo, nombre con el que se ha bautizado al programa, los usuarios del 664 sólo tendrán que insertar la cara dos del disco maestro o de una copia y teclear ICPM.

Los usuarios del 6128 deberán, forzosamente, sacar una copia del disco maestro número dos, insertarlo en la unidad de disco por la ca-



ra uno o dos, en función de que se desee trabajar con el Dr. Logo plus o con el 2.0 respectivamente, y teclear ICPM. Una vez que hemos entrado en el CPM habremos de teclear:

SUBMIT LOGO2

o bien:

SUBMIT LOGO3

La versión plus es idéntica a la dos, salvo que en ella se han incluido un cierto número de instrucciones, sobre todo en lo que concierne a la gestión de periféricos, impresora y disco. Un nuevo detalle que es de agradecer, es la inclusión de una instrucción que permite grabar la pantalla, gracias a lo cual podremos almacenar y conservar esas maravillosas y sencillas pantallas que realizaremos con nuestra tortuga.

Ambas implementaciones del Logo son bastante completas, el editor es agradable de utilizar y junto a todas las instrucciones que permiten mover la tortuga, pintar y demás, encontraremos todas las instrucciones necesarias para el manejo de listas con bastantes comodidades.

EL EDITOR

El editor del Logo es muy parecido al que se suministra con el Basic. Siempre podremos movernos con las teclas del cursor por la línea que estemos editando. Pero, ¿A qué esperamos?, escribamos nuestro primer procedimiento.

Lo primero que haremos será comunicar al ordenador que vamos a realizar un procedimiento, la forma que tenemos de realizar esto es comenzando por TO, continuando por el nombre del procedimiento.

TO ESE

recuerde teclear ENTER al final de cada línea.

RT 90

Esta instrucción le informa a la tortuga de que debe girar 90 grados.

FD 75

avanza 75

LT 90

gira 90 a la izquierda

FD 75

avanza 100

RT 90

FD 75

RT 90

FD 75

END

Esta última palabra le informa al ordenador de que el procedimiento ya se ha acabado.

Ahora teclee:

CS

Al pulsar ENTER la pantalla quedará en blanco y dividida en dos ventanas. Una para gráficos, la superior y otra para texto, la inferior.

Ahora el momento de la verdad, teclee:

ESE

Si no hay error el resultado lo tendrá en la pantalla.

En caso de que haya habido error y aunque no lo haya habido escriba:

ED «ESE»

Ahora tendrá en pantalla el procedimiento para poder modificarlo a placer. Con las teclas del cursor para moverse por toda su extensión o borrando con las teclas CLR y DEL, de forma similar a como lo hace en Basic, y entre FD 100 y END, cree una nueva línea:

RT 21

ES

La idea es hacer el primer procedimiento recursivo, pulse la tecla COPY para dar por terminado lo edición y teclee:

CS

ESE

El programa ahora no se detendrá. Cuando se haya cansado pulse la tecla ESC para detener la ejecución.

PROCEDIMIENTOS QUE LLAMAN A PROCEDIMIENTOS

Nuestro CPC en estos momentos, conoce lo que tiene que hacer cada vez que le digamos ESE. Suprima la última corrección que hemos hecho al programa, es decir, suprima lo antepenúltima y penúltima línea. Ya sabe, edite con:

ED "ESE"

Corrija, y para finalizar pulse COPY.

Creemos un nuevo procedimiento:

TO TRESES:

ESE

ESE

ESE

END

Acabamos de crear un nuevo procedimiento en el que utilizamos uno anterior que nuestro ordenador ya conoció.

NO TODO ES UNA TORTUGA

De todos modos emplear el LOGO y quedarse en la TORTUGA sería algo más que absurdo. Poco a poco intentaremos llegar al proceso de listas, utilizaremos los discos y la impresora, en resumen intentaremos conocer a

fondo este maravilloso lenguaje. Pero por ahora ya es suficiente, seguidamente le ofrecemos unos cuantos programas que le permitirán programar utilizando las órdenes que hoy hemos visto.

TO MASLOGO
ESPERAR ANALOGO
END.



```

to rombo      fd 10
rt 30         rt 120
fd 40         fd 10
rt 120        end
fd 40         to tapa
rt 60         lt 30
fd 40         fd 50
rt 120        lt 110
fd 40         fd 10
end           lt 30
to roseton   fd 50
rombo         end
rombo         to caj
rombo         fachada
rombo         costado
rombo         tape
rombo         end
rombo         to imposibl
rombo         cs
rombo         caja
rombo         raja
to vidriera  caja
roseton      caja
fd 40         caja
roseton      end
roseton      *
fd 40         ?cs
roseton      ? aja
fd 40         imposibl
roseton      to imposibl
end          caja
?             caja
?c            caja
?rombo       caja
?cs           end
?roseton    ?cs
?rs           ?imposible
?vidriera   to msiimposible
to fachada  fs
fd 50         cs
rt 90         impossibl
fd 50         rt 90
rt 90         impossibl
fd 50         rt 90
rt 90         impossibl
fd 50         rt 90
end          impossibl
to costado  nd
rt 30         ?
fd 35         ?mimposible
fd 60         ?mimposible

```

AMSTRAD

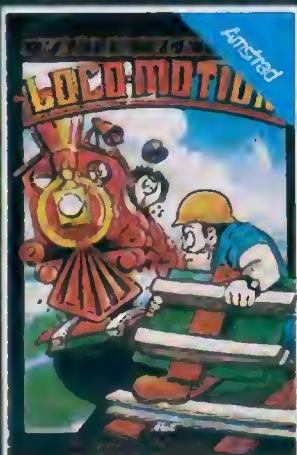
¡NO HAY COMPETENCIA POSIBLE!

en Calidad / Precio

MISTERTRONIC



COMMODORE - SPECTRUM - AMSTRAD



SPECTRUM - AMSTRAD

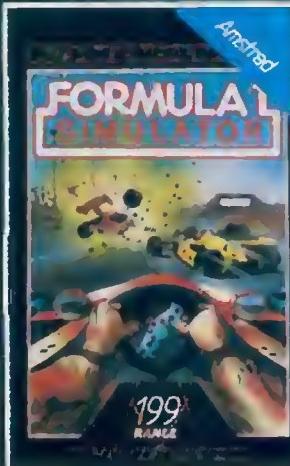
LOCOMOTION

La carrera más difícil de Europa, en la que 10 países europeos intentarán con piedras, troncos y coches, hacerte imposible la llegada. SUERTE ¡La necesitarás!

750
pts.

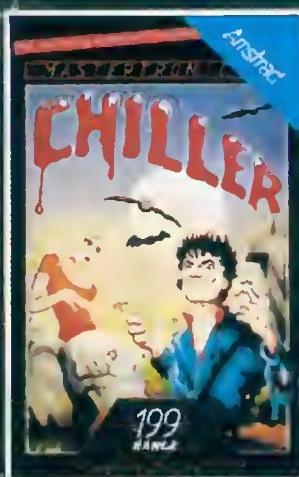


DRO SOFT



SPECTRUM
AMSTRAD
MSX

AMSTRAD - MSX



CHILLER

En una fría noche de Luna llena intentarás salvar a tu chica enfrentándote a cadáveres vivientes, arañas, espectros y murciélagos.

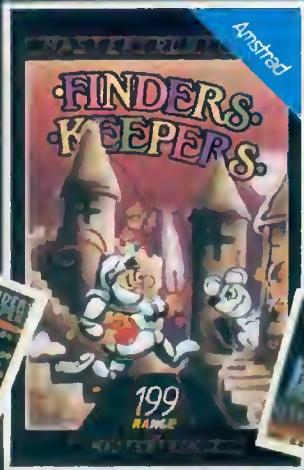
¡Anima y recoge todas las cruces que puedas!



FORMULA 1

El juego de competición de mayor realismo, con los 10 circuitos más famosos SILVERSTONE, MONACO, MONZA... etc.

¡3, 2, 1... Adelante!



COMMODORE
SPECTRUM
AMSTRAD
MSX



FINDERS KEEPERS

Demuestra que eres un caballero ofreciendo a tu dama regalos exóticos y duramente conseguidos.

Pero claro, es posible que prefieras quedar como un cerdo pero hacerte rico escapándote con la pasta.

Licencia exclusiva para ESPAÑA DRO SOFT
Fundadores, 3 - 28028 MADRID
Tels. 255 45 00/09

ENCUENTRALO
EN LA DIVISIÓN
Online DE

GALERIAS



UN RATON CASERO

Ya sabemos cómo usar al 100 por 100 los gráficos del Amstrad. Conocemos perfectamente cómo crear dibujos absolutos y relativos, líneas rectas y curvas y llenar superficies. Ahora bien, ¿cómo aplicar esto a algo útil? AMSTRAD Semanal ha intentado presentar una alternativa, entre las muchas que existen, con los programas de este artículo de Primeros Pasos. ¿Os suena el concepto de ratón, iconos y ventanas?

F. L. Frontán



emos recorrido ya un largo camino viendo todos las posibilidades gráficas de nuestro Amstrad. Siguiendo en esta línea vamos a intentar unir varias figuras y formas para componer un conjunto multicolor que quizás sea propuesta, quién sabe, de interés artístico general. **¡Manos a la obra!**

Vamos a analizar detalladamente lo que queremos hacer.

Ya conocemos alguna de las instrucciones necesarias para decirle a su ordenador que por un momento se convierte en artista. Dibujar en la pantalla cualquier forma más o menos geométrica no debe representar yo ningún problema para usted, ¿no es así?

Pues nuestra idea ahora es conseguir que nos aparezcan una serie de figuras que hemos establecido ya previamente, y que lo hagan, además, en cualquier punto de la pantalla. Y luego, si es posible, daremos color a su interior.

Elección de formas gráficas

Bueno, vamos elegir las figuras. Para no liarnos mucho vamos a seleccionar varios que no sean excesivamente complicadas pero que cumplan con la finalidad de esta sección que es ir cogiendo los conceptos necesarios para tener una amplia «base» informática que nos permita afrontar problemas mucho más complejos.

Vamos allá. Qué le parecen las siguientes: un círculo, un rectángulo, un cuadrado, un triángulo, la línea recta y, como antes dijimos, tener además la posibilidad de llenar el interior de la figura que dibujemos. ¡Vaya pensando en una serie de rutinas que nos tracen cada uno de estos dibujos!

Podemos hacerlo a base de un menú tradicional, sobre el que elegimos lo que queremos hacer, y luego nos bastaría con elegir el punto de la pantalla donde queremos colocar la forma geométrica.

Pero intentemos ser un poco más originales. Primeramente haremos aparecer en nuestra monitor una representación visual de cada una de las cosas que vamos a poder hacer. Es algo parecido a un menú gráfico.

En nuestro caso no es suficiente con que nos aparezcan unos rectángulos conteniendo la palabra que indique la acción a realizar. Por ejemplo, TRIANGULO será la zona que nos referenciará el poder dibujar un «triángulo» en un punto de la pantalla.

Menú de iconos

Además vamos a intentar conseguir un método para que en vez de seleccionar una opción mediante un menú clásico, podamos hacerlo simplemente señalándola en pantalla con un cursor, o flecha, y que con esto el ordenador ya sepa lo que tiene que hacer.

Todo lo que necesitamos es mover este cursor por la pantalla, mediante los teclados de movimiento del cursor, ¡claro! y hacer que apunte, o se coloque, encima de una de las opciones. Cuando pulsemos la tecla RETURN la máquina sobrá si queremos dibujar un rectángulo, trazar una recta o llenar de color rojo un círculo y nos mandará a la rutina encargada de realizar este trabajo.

Este es algo parecido a lo que se conoce por el concepto de ICONO, es decir, representar una acción procesable mediante una representación gráfica.

¿Comenzamos?

Como nos declaramos partidarios de seguir en programación una técnica modular, o sea, a base de pequeñas subrutinas que hagan cosas muy concretas, manejadas y seleccionadas a través de un programa principal, es el momento de aplicarlo a nuestro problema.

¿Cuál sería el programa que «diríja» toda esta orquesta? ¿Qué tendrá que hacer?



Primeros PASOS

Veamos. Ha de ser un programa que primero nos inicialice una serie de variables que necesitan tener un cierto valor al comienzo del programa.

A continuación nos dibujará la pantalla que nosotros hayamos diseñado. De momento no nos importa cuál va a ser su forma y contenido. Nos basta con saber que los gráficos de las opciones van a ocupar la parte de la izquierda hasta una determinada coordenada horizontal.

Una vez dibujada la pantalla comenzará a repetirse continuamente un ciclo en el que primero se mueve el cursor y después, si ha entrado en la zona de las opciones, compruebe si hemos elegido una de ellas.

¿Un poco abrumado por la charla anterior? Seguro que analizando el organigrama de la figura 1 verá todo mucho más claro.

Y de ahí a traducirlo a instrucciones Basic sólo hay un pequeño paso. O, ¿no es así? El programa uno nos demuestra que esto no es tan complicado como parece a primera vista.

Programa uno

Simplemente está compuesto por unas instrucciones que nos fijan el punto dónde estará colocado el cursor al principio del programa —líneas 40 y 50— y el color con el que comenzamos a trazar los gráficos —línea 60.

Después hay una llamada a la subrutina que nos dibujará la pantalla y que colocaremos a partir de la línea 11000.

En la 80 moveremos el cursor, si lo creemos necesario, mediante un salto a la subrutina de movimiento que empezará en la línea 1000.

Sólo nos queda analizar si podemos elegir una opción —línea 90— cuando estemos en una coordenada horizontal menor que el límite propuesto para la zona de «íconos».

Y volveremos a repetir el ciclo mediante un salto a la línea 80. ¿Verdad que no es muy complicado?

Nuestro próximo trabajo consiste en ir dando forma a cada uno de las subrutinas que vamos a manejar. ¿Empezamos con la que nos dibuja la pantalla? Va a ser muy sencillito. Veo el programa dos.

Programa dos

Lo primero que hace es recuadrar los límites de la pantalla y delimitar la zona donde vamos a colocar el menú gráfico —líneas 11000 a 11130.

Después se coloca cada uno de los textos dentro de su correspondiente zona de «ícono» mediante el bucle FOR... NEXT de las líneas 11140 a 11180 que va leyendo los DATAs de la línea 11260.

A continuación rellenaremos con los cuatro colores existentes en el Modo 1 la zona del menú destinada a seleccionar el color de los gráficos —líneas 11190 a 11240.

Y ya está. Eche un vistazo al organigrama de la figura 2 y analice más detenidamente el proceso lógico que hemos seguido para llegar a codificar el programa dos.

Si seguimos el diagrama de flujo del programa principal, la siguiente acción a realizar es mover el cursor según las teclas que hayamos pulsado.

La manera de hacerlo es siguiendo el gráfico de la figura 3.

Realizamos el siguiente proceso.

— Borrar el cursor de la última posición donde lo hemos dibujado, o iluminarlo con el mismo color del fondo de la pantalla.

— Analizar si hemos pulsado alguna tecla e incrementar o decrementar las coordenadas del cursor dependiendo de dicho análisis.

— Asegurarse de que el cursor no se salga de los límites establecidos para el dibujo.

— Anotar el color del punto situado en las nuevas coordenadas calculadas.

— Iluminar el cursor en su nuevo situación.

— Actualizar las antiguas coordenadas del cursor.

— Lo último que nos queda es devolver control al programa principal.

¿Qué le ha parecido? Si sigue paso a paso este organigrama y introduce a instrucciones Basic cada uno de los apartados, seguro que obtendrá un programa semejante al tres.

Programa tres

Los únicos novedades que encontramos están en las líneas 1030 a 1060 y en la línea 1090.

INKEY (N) es una función que nos devuelve el valor 0 si la tecla que se corresponda con el número «N» ha sido pulsada. Cuando no la hayamos presionado nos devolverá el valor -1.

En nuestro caso las teclas asociadas a los números 0, 1, 2, y 8 son precisamente las de movimiento del cursor. Observe cómo se incrementa la coordenada correspondiente cuando hemos pulsado una de estas teclas —o cuando la función INKEY nos devuelve un cero.

La otra instrucción desconocida es TEST (X, Y). Es una función que nos devuelve el valor de la tinta con la que hemos iluminado el punto situado en las coordenadas gráficas x, y.

Nosotros la utilizamos en esta ocasión para poder borrar el punto antes de calcular sus nuevas coordenadas y así evitar que el movimiento vaya dejando estela.

Las demás líneas ya son conocidas. En la línea 1020 devolvemos al cursor el color del fondo. Con la 1070 y 1080 evitamos que el cursor se salga de la pantalla.

En la 1100 iluminamos el punto en su nueva posición y con las 1110 y 1120 actualizamos las coordenadas.

Y con esto terminamos, devolvemos el control al programa principal y punto.

Continuemos. Despues pasamos a analizar si estamos en condiciones de seleccionar una determinada opción. Para ello el cursor debe estar en la zona del menú gráfico, o su coordenada x ser menor que 192 como se observa en la línea 90 del programa principal. Vemos la rutina que nos permite hacer la elección.

Cuando estemos situados en el lugar adecuado (*menú*) pulsamos la tecla RETURN para anotar la acción que vamos a realizar.

Una vez hecha la selección moveremos el cursor a la zona de dibujo y allí trazaremos la figura elegida.

¿Qué hemos de hacer para que esto ocurra así? Siga el proceso de la figura 4 y lo verá claramente.

Daremos los siguientes pasos:

— El cursor se estará moviendo hasta que pulsemos la tecla RETURN.

— En ese momento anotaremos la acción elegida dependiendo de la coordenada vertical del punto donde esté situada y soltará el programa a la subrutina que la ejecuta.

Con el programa cuatro comprobaremos cómo tendríamos que codificar todo esto para que funcione tal como queremos.

Programa cuatro

Hacemos que el cursor se esté moviendo mediante el bucle WHILE... WEND de la línea 2020 a 2040. INKEY(18) detecta el momento en que pulsamos la tecla RETURN.

Una vez pulsada, y por tanto seleccionada la opción, saltaremos a la subrutina que corresponda después de evaluar las condiciones reflejadas en las líneas 2050 a 2120.

Y volveremos al programa principal al encontrarnos con la instrucción RETURN.

¿Qué nos queda para cumplir el recorrido? Echando un vistazo a la línea 100 del programa 1 (*el principal*) observamos que se vuelve a la línea 80 y repetimos el ciclo.

Ya hemos visto el funcionamiento general de todo este invento. Es conveniente ahora bajar a analizar cómo vamos a realizar cada una de las opciones del menú. Pasamos a ello.

Como ya hemos visto la manera de dibujar alguna figura geométrica en algunos artículos anteriores, sólo nos detendremos en fijarnos en la manera de determinar el centro y

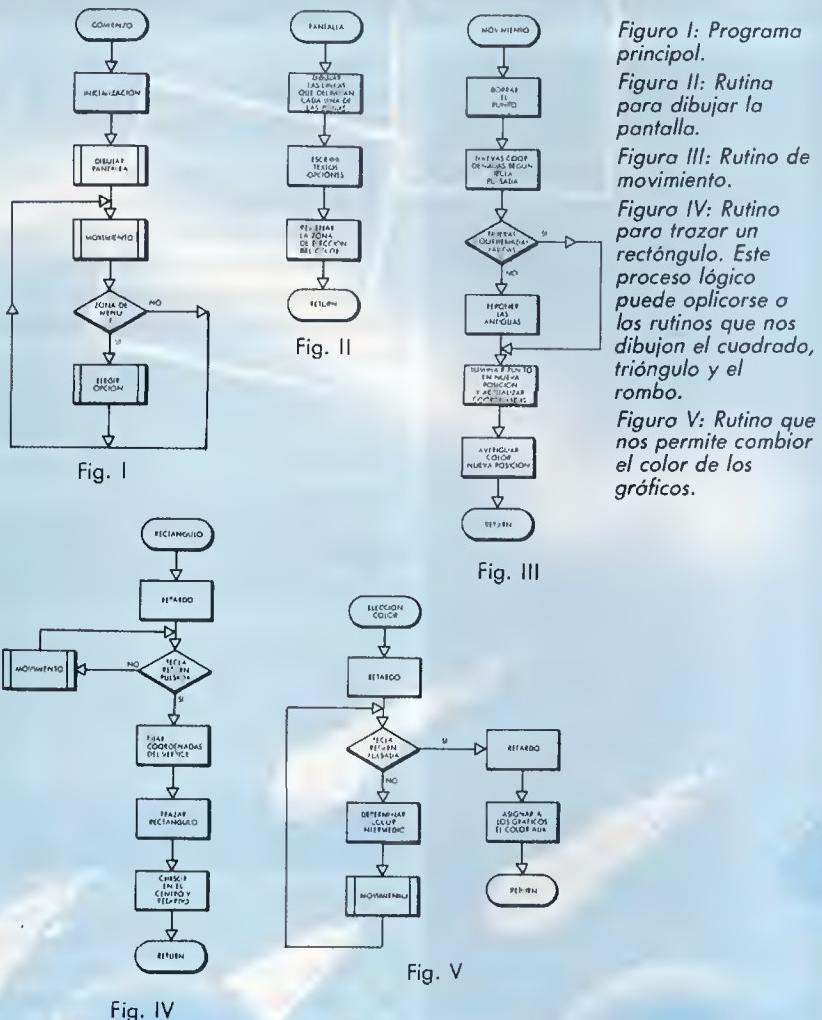


Figura I: Programa principal.

Figura II: Rutina para dibujar la pantalla.

Figura III: Rutina de movimiento.

Figura IV: Rutina para trazar un rectángulo. Este proceso lógico puede aplicarse a los rutinos que nos dibujan el cuadrado, triángulo y el rombo.

Figura V: Rutina que nos permite combinar el color de los gráficos.

el radio en el caso de la circunferencia o el punto de origen y el final de la recta o un vértice en el caso de las demás figuras.

Para fijar estos puntos recurrimos al empleo de un bucle WHILE... WEND que detecte si hemos pulsado la tecla RETURN o no. Mientras no la presionemos, el cursor seguirá moviéndose.

Una vez pulsada, anotamos las coordenadas del punto donde nos encontramos y se lo asociaremos a uno de los casos comentados anteriormente: el centro de la circunferencia, por ejemplo.

Cuando tengamos determinados todos los puntos necesarios, la rutina correspondiente le dibujará la figura elegida.

Observe que en caso de necesitar detectar más de una pulsación de la tecla RETURN, metemos por medio un bucle retardador semejante a:

10 FOR retardo = 1 TO 1000: NEXT retardo

ya que si no, debido a la rapidez con que se ejecuta no nos daría tiempo a soltar la tecla en el momento preciso.

Podemos aplicar todo lo anterior a la hora de fijar el color con el que vamos a dibujar una figura o el que vamos a utilizar para rellenar otra. Recuerde, para cualquier elección ¡PULSE RETURN!

Por si acaso le ha quedado algo oscuro, le

remitimos a los organigramas y programas de cada una de estas subrutinas. Obsérvelos detenidamente y verá cómo se le aclaran las posibles dudas.

Y antes de terminar, un respiro. El proceso a seguir para dibujar una figura es:

— Elegir el color con el que vamos a dibujar los gráficos en caso de que no coincida con el cursor. Para ello le colocamos encima del color elegido y pulsamos RETURN si es correcto.

— Seleccionar una opción llevando el cursor encima de la zona del menú elegida y pulsando la tecla RETURN a continuación.

— Fijar los puntos que necesitemos en la zona de dibujo de la pantalla para trazar la figura elegida (*dos en el caso de la recta y la circunferencia y uno en los demás casos*).

— Si queremos llenar de color una figura colocamos el cursor en la opción RELLENAR, pulsando después RETURN. A continuación seleccionamos de una forma análoga el mismo color que hemos utilizado para dibujar el gráfico. Después movemos el cursor dentro de la figura y si pulsamos de nuevo RETURN se nos llenará de color.

Bueno. Es todo por esta semana. Si quiere ver en funcionamiento este «invento» puede teclear todos estos programas seguidos o bien mezclarlos con MERGE si lo escribe de uno en uno. La elección es suya. Hasta pronto.

PROGRAMAS

```

10 REM PROGRAMA I
20 REM PROGRAMA PRINCIPAL
30 DEG
40 LET x=420
50 LET y=200
60 color=1
70 GOSUB 11000:REM PANTALLA
80 GOSUB 1000:REM MOVIMIENTO
90 IF x<192 THEN GOSUB 2000:REM ELE
GIR OPCION
100 GOTO 80

11000 REM PROGRAMA II
11010 REM CREACION PANTALLA
11020 CLS
11030 PLOT 0,16,1
11040 DRAW 639,16
11050 DRAW 639,399
11060 DRAW 0,399
11070 DRAW 0,16
11080 MOVE 191,16
11090 DRAW 191,399
11100 FOR linea=64 TO 399 STEP 4B
11110 MOVE 190,linea
11120 DRAW 0,linea
11130 NEXT linea
11140 FOR texto=2 TO 20 STEP 3
11150 READ opcion$
11160 LOCATE 2,TEXT0
11170 PRINT opcion$
11180 NEXT texto
11190 FOR opcolor=0 TO 3
11200 FOR columna=2 TO 47
11210 MOVE 46*opcolor+columna,1B
11220 DRAW 46*opcolor+columna,63,op
color
11230 NEXT columna
11240 NEXT opcolor
11250 DATA " CIRCULO","RECTANGULO",
" CUADRADO","TRIANGULO"," RECTA",
" ROMBO"," RELLENAR"
11260 RETURN

1000 REM PROGRAMA III
1010 REM RUTINA DE MOVIMIENTO
1020 PLOT x,y,color1
1030 IF INKEY(0)=0 THEN y=y+2
1040 IF INKEY(2)=0 THEN y=y-2
1050 IF INKEY(1)=0 THEN x=x+2
1060 IF INKEY(B)=0 THEN x=x-2
1070 IF x<2 OR x>637 THEN x=xant
1080 IF y<18 OR y>397 THEN y=yant
1090 color1=TEST(x,y)
1100 PLOT x,y,color1
1110 xant=x
1120 yant=y
1130 RETURN

2000 REM PROGRAMA IV
2010 REM RUTINA ELEGIR OPCION
2020 WHILE INKEY(1B)<>0
2030 GOSUB 1000
2040 WEND
2050 IF y>353 THEN GOSUB 3000:RETUR
N
2060 IF y>305 AND y<351 THEN GOSUB
4000:RETURN
2070 IF y>257 AND y<303 THEN GOSUB
5000:RETURN
2080 IF y>209 AND y<255 THEN GOSUB
6000:RETURN
2090 IF y>161 AND y<207 THEN GOSUB
7000:RETURN
2100 IF y>113 AND y<159 THEN GOSUB
8000:RETURN
2110 IF y>65 AND y<111 THEN GOSUB 9
000:RETURN
2120 IF y<63 THEN GOSUB 10000:RETUR
N

3000 REM PROGRAMA V
3010 REM RUTINA DIBUJAR CIRCULO
3020 LOCATE 1,25:PRINT"CIRCULO ";
3030 FOR retardo=1 TO 1000:NEXT
3040 WHILE INKEY(1B)<>0 OR x<192
3050 GOSUB 1000
3060 WEND
3070 xcorigen=x
3080 yorigen=y
3090 PRINT"DE CENTRO";xcorigen;",";y

```

```

origen;
3100 FOR retardo=1 TO 1000:NEXT
3110 WHILE INKEY(1B)<>0
3120 GOSUB 1000
3130 WEND
3140 radio=SQR((x-xorigen)^2+(y-yor
igen)^2)
3150 PRINT"RADIO";INT(radio)
3160 PLOT xcorigen+radio,yorigen,col
or
3170 FOR angulo=0 TO 360
3180 xcir=xorigen+radio*COS(angulo)
3190 ycir=yorigen+radio*SIN(angulo)
3200 IF xcir<192 THEN xcir=192
3210 IF ycir<18 THEN ycir=18
3220 DRAW xcir,ycir
3230 NEXT angulo
3240 x=xorigen
3250 y=yorigen
3260 LOCATE 1,25:PRINT"
3270 RETURN

```

```

4000 REM PROGRAMA VI
4010 REM RUTINA DIBUJAR RECTANGULO
4020 LOCATE 1,25:PRINT"RECTANGULO";
4030 FOR retardo=1 TO 1000:NEXT ret
ardo
4040 WHILE INKEY(1B)<>0 OR x<192
4050 GOSUB 1000
4060 WEND
4070 PRINT"(vertice";x;";";y;")"
4080 DRAWR 200,0
4090 DRAWR 0,100
4100 DRAWR -200,0
4110 DRAWR 0,-100
4120 xnueva=x+100
4130 IF xnueva<637 THEN x=xnueva EL
SE x=637
4140 ynueva=y+50
4150 IF ynueva<397 THEN y=ynueva EL
SE y=397
4160 FOR retardo=1 TO 1000:NEXT ret
ardo
4170 LOCATE 1,25:PRINT"
4180 RETURN

```

```

5000 REM PROGRAMA VII
5010 REM RUTINA DIBUJAR CUADRADO
5020 LOCATE 1,25:PRINT"CUADRADO";
5030 FOR retardo=1 TO 1000:NEXT ret
ardo
5040 WHILE INKEY(1B)<>0 OR x<192
5050 GOSUB 1000
5060 WEND
5070 PRINT"(vertice";x;";";y;")"
5080 DRAWR 150,0
5090 DRAWR 0,150
5100 DRAWR -150,0
5110 DRAWR 0,-150
5120 xnueva=x+75
5130 IF xnueva<637 THEN x=xnueva EL
SE x=637
5140 ynueva=y+75
5150 IF ynueva<397 THEN y=ynueva EL
SE y=397
5160 FOR retardo=1 TO 1000:NEXT ret
ardo
5170 LOCATE 1,25:PRINT"
5180 RETURN

```

```

6000 REM PROGRAMA VIII
6010 REM RUTINA DIBUJAR TRIANGULO
6020 LOCATE 1,25:PRINT"TRIANGULO";
6030 FOR retardo=1 TO 1000:NEXT ret
ardo
6040 WHILE INKEY(1B)<>0 OR x<192
6050 GOSUB 1000
6060 WEND
6070 PRINT"(vertice";x;";";y;")"
6080 DRAWR 200,0
6090 DRAWR -200*COS(60),200*SIN(60)
6100 DRAWR -200*COS(60),-200*SIN(60)
)
6110 xnueva=x+(200*COS(60))
6120 IF xnueva<637 THEN x=xnueva EL
SE x=637
6130 ynueva=y+(200*SIN(60)/2)
6140 IF ynueva<397 THEN y=ynueva EL
SE y=397

```

```

6150 FOR retardo=1 TO 1000:NEXT ret
ardo
6160 LOCATE 1,25:PRINT"
6170 RETURN

```

```

7000 REM PROGRAMA IX
7010 REM RUTINA TRAZAR RECTA
7020 LOCATE 1,25:PRINT"RECTA";
7030 FOR retardo=1 TO 1000:NEXT ret
ardo
7040 WHILE INKEY(1B)<>0 OR x<192
7050 GOSUB 1000
7060 WEND
7070 PRINT"(origen";x;";";y;")"
7080 xcorigen=x
7090 yorigen=y
7100 FOR retardo=1 TO 1000:NEXT
7110 WHILE INKEY(1B)<>0
7120 GOSUB 1000
7130 WEND
7140 PRINT"-fin";x;";";y;")"
7150 PLOT xcorigen,yorigen
7160 DRAW x,y,color
7170 FOR retardo=1 TO 1000:NEXT
7180 LOCATE 1,25:PRINT"
7190 RETURN

```

```

8000 REM PROGRAMA X
8010 REM RUTINA DIBUJAR ROMBO
8020 LOCATE 1,25:PRINT"ROMBO";
8030 FOR retardo=1 TO 1000:NEXT ret
ardo
8040 WHILE INKEY(1B)<>0 OR x<192
8050 GOSUB 1000
8060 WEND
8070 PRINT"(vertice";x;";";y;")"
8080 DRAWR 100*COS(60),100*SIN(60)
8090 DRAWR 100*COS(60),-100*SIN(60)
8100 DRAWR -100*COS(60),-100*SIN(60)
)
8110 DRAWR -100*COS(60),100*SIN(60)
8120 xnueva=x+(100*COS(60))
8130 IF xnueva<637 THEN x=xnueva EL
SE x=637
8140 FOR retardo=1 TO 1000:NEXT ret
ardo
8150 LOCATE 1,25:PRINT"
8160 RETURN

```

```

9000 REM PROGRAMA XI
9010 REM RELLENAR FIGURA
9020 LOCATE 1,25:PRINT"ELIGE COLOR"
9030 GOSUB 10000
9040 LOCATE 1,25:PRINT"
9050 FOR retardo=1 TO 1000:NEXT ret
ardo
9060 WHILE x<192
9070 GOSUB 1000
9080 WEND
9090 WHILE INKEY(1B)<>0
9100 GOSUB 1000
9110 WEND
9120 MOVE x+2,y
9130 FILL color
9140 RETURN

```

```

10000 REM PROGRAMA XII
10010 REM ELECCION DE COLOR
10020 FOR retardo=1 TO 1000:NEXT re
tardo
10030 WHILE INKEY(1B)<>0 AND x<192
10040 IF x>144 THEN coloraux=3
10050 IF x>97 AND x<144 THEN color
aux=2
10060 IF x>50 AND x<97 THEN colorau
x=1
10070 IF x<50 THEN coloraux=0
10080 GOSUB 1000
10090 LOCATE 1,25:PRINT"COLOR";colo
raux;" PULSA RETURN SI ES EL ELEGID
O"
10100 WEND
10110 FOR retardo=1 TO 1000:NEXT re
tardo
10120 LOCATE 1,25:PRINT"

```

```

10130 color=coloraux
10140 RETURN

```

ANALISIS



n esta ocasión analizaremos un programa basado en el cálculo de errores cometidos por el individuo al realizar la medición.

El desglose de las líneas es el siguiente:

10-40 Estas sentencias REM dan información a las personas que leen el listado. Están separadas del resto del programa y son completamente ignoradas por el ordenador. Estos REMs se pueden sustituir por la coma sendilla (').

50 REM que informa del comienzo del programa y de la entrada de los datos de la medición.

60 Limpia la pantalla.

70 Se crea la ventana #1 con el comando WINDOW, situada y dimensionada según los parámetros indicados.

80 Pregunta el número de medidas realizadas y las asigna a la variable N.

90 Especifica con la orden IF... THEN... que si se realiza un número de medidas menor que 2, la ejecución del programa vuelve a la línea 50, ya que es ilógico hacer un cálculo de errores tomando menos de 2 medidas.

100 Con el comando DIM se crea una matriz de medidas A y se especifica cuál va a ser su número de elementos con la variable N.

CALCULO DE ERRORES

Cuando realizamos una medida de cualquier magnitud, la medición no es completamente exacta, hay un pequeño error producido por el aparato con el que se realiza la medida o por el individuo.

110 Para más claridad del programa se pone el contador S (sumatorio de medidas) a cero, aunque el Amstrad lo da ya por realizado.

120 Escribe en pantalla el texto «MEDI-DAS»:

130 Comienzo un bucle FOR... NEXT cuya variable de control es I y que tiene un número de iteraciones máximo igual a la variable N (número de medidas).

140 Borra la ventana #1.

150 Imprime en ella el texto con el número de la medida a introducir.

160 Pregunta las medidas realizadas, ordenadas según la variable I. A cada una de ellas se le asigna una posición dentro de la matriz, mediante A(I).

170 Escribe en pantalla (*a continuación del texto impreso en la línea 120*) la lista de las medidas, ordenadas y separadas por un guion.

180 En cada iteración se suma a la variable S las distintas medidas realizadas.

190 Finaliza el ciclo FOR...NEXT.

200 Borra la ventana #1.

210 REM que informa del comienzo del cálculo del valor medio.

220 Calcula la variable VM (valor medio) mediante el cociente entre el sumatorio de medidas y el número de las realizadas. VM=S/N.

230 Representación en pantalla de la solución encontrada para el valor medio. Esta viene realizada utilizando la plantilla de formato contenida en el comando DEC\$. Esto es: 6 posiciones para contener un número entero con un máximo de 6 cifras, y 4 posiciones para contener 4 decimales.

240 REM que informa del comienzo del cálculo de la desviación típica.

250 Para mayor claridad del programa se pone la variable D a cero.

260 Inicia un nuevo bucle FOR... NEXT igual que el anterior.

270 En cada iteración se suma a la variable D el cuadrado de la resta entre cada medida ordenada y el valor medio D=D+(A(I)-VM)/2.

280 Acaba el ciclo FOR... NEXT.

290 Calcula la variable DT (desviación típica) mediante la raíz cuadrada del cociente entre la variable D y el número de medidas realizadas menos 1. DT=SQR (D/(N-1)).

300 Se imprime en pantalla la solución a la desviación típica. El resultado está representado mediante el comando DEC\$, igual que en la línea 230.

310 REM que informa del comienzo del cálculo de error cuadrático medio.

320 Calcula la variable ECM (error cuadrático medio) mediante el cociente entre la desviación típica y la raíz cuadrada del número de medidas. ECM=DT/SQR (N).

330 Se escribe en pantalla la solución al error cuadrático medio. El resultado está representado igual que en líneas anteriores.

340 Impresión en pantalla de una advertencia y consejo para realizar el error total.

350 Llamada a la rutina del FIRMWARE, que detiene la ejecución del programa hasta que se pulse una tecla. Esta rutina comienza en la dirección hexadecimal BB06.

360 Esta instrucción manda que la ejecución del programa salte a la línea 50.



Para que tus dedos
no realicen el trabajo duro, M+AMS
TRAD lo hace por ti. Todos los listados que incluyen
este logotipo se encuentran a tu disposición en un cas-
sette mensual, solicitalo.

SI BUSCAS LO MEJOR



Software

LO TIENE

PING PONG



The officially endorsed version
of the No.1 arcade game by

Konami®



DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE C/. STA. ENGRACIA, 17 - 28010 MADRID, TFNO. (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, N.º 10. TFNO. (93) 432 07 31



N.º 32

N.º 33

N.º 34

N.º 35

N.º 36

N.º 37

N.º 38

N.º 39



Software



Nombre

Apellidos

Domicilio Tel.

Localidad Provincia

*Todo el Software comercializado por ERBE,
durante un año, para tu AMSTRAD,*

**¡PUEDE SER TUYO
COMPLETAMENTE
GRATIS!**

BASES

* Cada semana, durante los dos próximos meses, aparecerá dentro de las páginas de la revista un cupón como el de muestra, que deberás recortar y pegar en el lugar correspondiente de la página 16 de este mismo número.

* Asimismo, en todos los programas distribuidos por ERBE SOFTWARE, puedes encontrar los logotipos de U.S. GOLD y de la propia ERBE. Cualquiera de los dos tipos te valdrá para rellenar los dos huecos de la página mencionada.

* Una vez rellena la página con los 8 símbolos de AMSTRAD SEMANAL y los 2 de ERBE o U.S. GOLD (más los de muestra que ya están colocados)

envianosla a HOBBY PRESS, apartado de correos 232 de ALCOBENDAS (MADRID), sin olvidar incluir, claramente expresados, todos tus datos personales en el espacio reservado al efecto.

* Todas las cartas recibidas antes del día 30 de junio de 1986, participarán en un sorteo que tendrá lugar el día 1 de julio en nuestra redacción. Los diez ganadores recibirán

COMPLETAMENTE GRATIS
y durante un año,
TODOS LOS PROGRAMAS que ERBE SOFTWARE distribuya para el AMSTRAD.

¡¡Participa en esta extraordinaria promoción!!

**No pierdas esta oportunidad
que te brindan ERBE
y AMSTRAD SEMANAL**

THREE WEEKS IN PARADISE

La exuberante vegetación de la jungla tropical, es el escenario de la última y más peligrosa aventura de la familia Week; caníbales, cocodrilos, leones, pirañas y demás peligros de la floresta, se interponen entre Wally y su familia.

La carrera de Wally ha sido fulgurante; lejos quedan aquellos tiempos en que era mecánico de automóviles en una pequeña ciudad.

Harto de grasa, tuercas y olor a gasolina, su cabeza se pone a pensar en formas más rápidas de hacer dinero.

Con ayuda del clan familiar y algunas amigos invitados, prepara un astuto plan, cuyo objetivo es el Banco Central de su ciudad, el cual dejar limpia de fondos.

Tras el audaz golpe, la familia continúa con su vida normal, no mostrando signos externos de su nueva situación económica. Por otro lado, las investigaciones policiales sobre el golpe continúan adelante, y el cerebro de la banda empieza a considerar si su situación es totalmente segura.

Consciente de que las técnicas de investigación criminal han avanzado mucho, y nada se escapa a los ojos de la ley, decide abandonar el país, y esconderse en un Estado que no tenga acuerdo de extradición con la Gran Bretaña.

Para no levantar sospechas, la familia se embarca en un crucero de placer al África negra, desapareciendo en uno de los puertos de escala, para dirigirse a una isla aparentemente deshabitada, en la cual permanecer a salvo del implacable brazo de la justicia.

Cuando los tres componentes de la familia se disponen a llevar una vida totalmente ecológica en contacto con la naturaleza, entran en escena los caníbales que habitan en la parte alta de la isla y de los cuales la familia no imaginaba ni siquiera su existencia.

El resultado de esta visita inesperada, es que Wilma y Herbert, respectivamente mujer e hijo de Wally, son atrapados y se encuentran en serio peligro de transformarse en una de las especialidades culinarias del cocinero de la tribu.

Wally, que no ha sido atrapado por los indígenas, pues en el momento del asalto se encontraba subido a un cocotero, tiene que liberar a su querida familia al precio que sea.

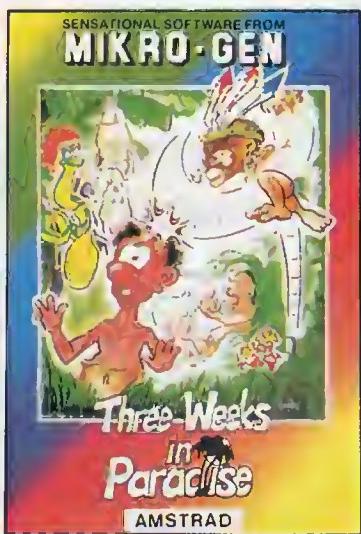
Para ello vagará incansablemente por la jungla, combatiendo a brazo partido contra leones, cocodrilos, indígenas, abejorros gigantes y demás moradores. Descenderá a las profundidades acuáticas, escalará árboles centenarios y no dejará un palmo de terreno sin explorar en busca de los suyos.



Un programa que sigue fielmente la línea de las anteriores aventuras de Wally; en un escenario tropical debemos desvelar la complicada trama que nos hará rescatar a los componentes de la familia.

Los gráficos que nos acompañan en el transcurso de la aventura, están realizados en el modo de menor resolución, que en detrimento de ésta, dota al programa de un colorido muy adecuado al escenario elegido; árboles, vegetación y demás elementos, siguen la pauta de los programas de la casa. Realizados con un diseño sencillo y muy descriptivo, dan vida a las distintas pantallas.

A nuestro paso, mientras reconocemos las zonas de la isla por las que podemos circular, encontramos multitud de objetos, cada uno útil para una sola misión; botellas, espadas, hachas, martillos etc. tienen solamente uso en un lugar determinado de la isla, el cual hay que descubrir a base de paciencia.



Compatible: CPC/464, CPC/664 y CPC/6128

	HORRIBLE
	UN ROLLO
	PASABLE
	BUENO
	MUY BUENO
	OBRA MAESTRA



Mister JOYSTICK

**Three Weeks
in Paradise**



A pesar de todo, completar el 100 por 100 de la aventura solamente está al alcance de los creadores del juego, pues esta nueva aventura de Wally, sigue en la línea de las anteriores, y todas sabemos lo que cuesta completar un Pyjamarama o un Evryone's a Wally, una misión imposible.

Un programa para verdaderos adictos, a los que esos juegos tan fáciles que se resuelven en dos sentadas, les dejan totalmente fríos.

En este tendrán que usar toda su astucia y perseverancia, encontrando en cada nuevo enfrentamiento con el programa, esas sutilezas que permiten encontrar sentido a ciertas objetos en determinadas pantallas, fuera de las cuales no sirven absolutamente para nada.

La trama de la aventura, está encadenada lógicamente, de manera



que las acciones a llevar a cabo, dependen unas del anterior cumplimiento de otras, constituyendo un proceso complicado, en el cual conseguir descubrir el uso que se debe dar a algunos de los objetos con que nos encontramos, es un gran logro.

Averiguar en cada nueva partida el uso de un objeto que no conocemos, debe ser la meta de nuestras investigaciones por la exuberante jungla tropical. Gracias a determinados objetos podremos descubrir pantallas de las que no teníamos ni noción y avanzando poco a poco en el uso de todos ellos, unas meses de práctica nos pondrán en condiciones de realizar infinidad de descubrimientos en las distintas zonas de la selva.



AMPLIACION DE MEMORIA DK'TRONICS

Los ordenadores Amstrad cada vez están causando un impacto mayor en todo el mundo. Esto se refleja, lógicamente, en todos los productos propios de todo computador, pero fabricados por otras personas: aumentan de número rápidamente.

Así ha ocurrido con el goloso tema de las ampliaciones de memoria, por las obvias ventajas que conlleva el disponer de más en tu ordenador.

Banco de pruebas analiza las más populares del mercado: las ampliaciones de DK'TRONICS.



P

ara todos aquellos que se quejaban de falta de memoria en su ordenador **Amstrad CPC**, la firma inglesa **DK'TRONICS** acaba de comercializar en España la ampliación de memoria que aumenta en 64 Kbytes nuestro ordenador.

Dk'Tronics se estrena en el mercado del ordenador **Amstrad** con una nueva gama de periféricos que fortalecen aún más las posibilidades de este ordenador. De la nueva gama de productos vamos a tratar en esta ocasión de la ampliación de 64 K compatible para los modelos CPC 464 y 664. Dicha ampliación consta de una parte de Software y otra de Hardware. La primera incluye los nuevos comandos RSX en soporte de cinta magnética con los cuales accederemos a los nuevos bancos de memoria. La segunda contiene los chips de RAM que aumentan la capacidad en cuatro nuevos bloques de 16 K cada uno y se conecta al ordenador a través del port trasero del «Floppy Disc», llevando en su parte posterior el bus de expansión para conectar otros periféricos como la Unidad de Disco.

El manual que acompaña a este kit consta de 22 páginas en el cual se explican claramente los nuevos comandos, así como el acceso a los diferentes bancos de memoria. Trae ejemplos de la utilización de estas instrucciones, aunque de momento vienen publicado en inglés, el distribuidor oficial en España nos ha asegurado que saldrán próximamente a la venta en castellano.

Posibilidad de utilización del CP/M PLUS

Una de las mejores aplicaciones de esta ampliación es la utilización del CP/M PLUS hasta ahora de uso exclusivo para el CPC-6128. Sin embargo, esta posibilidad no queda bien aclarada en el manual (*ton sólo nos hablo seis líneas en dos párrafos*). Uno de los defectos que adolecía el CP/M 2.2 era la imposibilidad de utilizar compiladores de otros lenguajes a un alto nivel, ya que los que hasta ahora se comercializan, tienen muy recortadas sus posibilidades. Por citar algunos ejemplos el compilador de Cobol-80 de Microsoft o el Fortran-80 de esta misma casa son ahora accesibles con los modelos 464 y 664.

Por otra parte toda la gama de programas de gestión que corre bajo CP/M PLUS está al alcance con esta nueva ampliación.



Puesta en marcha

La conexión del Hardware al bus de expansión se debe de realizar con el ordenador apagado para evitar posibles cortocircuitos entre sus diferentes pistas. Una vez realizado ésta, nos disponemos a cargar el Software que nos introduce los comandos de RSX y nos reconoce los nuevos bancos.

Una vez cargado el programa nos preguntará por la dirección de carga en la que se ubicará el programa. Esta dirección va desde la dirección 32768 a la 41811; en el caso de no especificar ninguna dirección y pulsar directamente la tecla de Return o Intro éste se cargará en la dirección más alta. Comprobaremos que en la pantalla nos sale un test de los bancos de memoria, indicándonos si hay algún error y en qué banco se encuentra. El total de bancos nuevos es de cuatro, de 16 K cada uno. El sistema viene aumentada con otros mensajes de error. Estos son:

Bad Bank Command: Se comete un error al poner un parámetro a si la variable no está presente donde debería estar situada.

Bank Unavailable: Si se intenta entrar en el banco que no está presente en el sistema.

Bad Bank Parameter: Hemos utilizado un parámetro inadecuado en un banco de memoria.

Bad Bank Address: La dirección pedida está fuera del rango del banco. Esta va desde la 0 hasta la 16383.

Value Invalid: La dirección del banco es más grande que la definida en el dato.

Bad Window Definition: El número de ventana es superior a siete, que es el máximo que permite. Ahora ya estamos preparados para programar contando con los bancos de memoria adicionales que le hemos incorporado. O bien ejecutar el CP/M PLUS con sus programas. Aunque antes hemos de ejecutar un comando nuevo: Emulate.

Nuevos comandos para el Amstrad

Doce son las nuevas instrucciones que nos aporta la ampliación. Algunas de estas instrucciones pueden llevar uno o varios parámetros, o ninguno según el caso. Una ventaja que hemos podido observar es la fácil asimilación de estas nuevas instrucciones. Para dar una pequeña idea de éstas y cómo funcionan podemos hacer cinco divisiones.

La primera división es la encargada del manejo de pantallas con los bancos; las encargadas de cargar y salvar las pantallas en este caso son: Loads y Saves respectivamente. Estos comandos pueden estar acompañados de varios parámetros:

Loads, [Banco], [Swap]
Saves, [Banco], [Swap]

El primer parámetro es obligatorio y varía desde 1 hasta 4, mientras que el segundo es opcional. El segundo valor debe de ser un 0 ó un 1 dependiendo si queremos activarlo o no. Si se omite este parámetro toma por defecto el valor 0.

Banco de PRUEBAS

Vamos ha utilizar estas nuevas sentencias en el primer programa para observar su ejecución y poder sacarle toda el partido a esta sentencia en nuestros programas.

La segunda división es la encargada del manejo de ventanas con los bancos de RAM. Son ocho las ventanas que podemos crear en los bancos de memoria numeradas del 0 al 7. Otros dos nuevos comandos se nos incorporan a nuestro Basic para cargar o salvar ventanas: Loadw y Savew respectivamente. Al igual que antes estos comandos pueden ir acompañados por varios parámetros, aunque no sean todos necesarios a la hora de emplearlos:

- Loadw, [Número Ventana], [Banco], [Dirección del Banco]
- Savew, [Número Ventana], [Banco], [Dirección del Banco]

El primer parámetro es obligatorio y nos indica cuál es el número de la ventana, número comprendido entre 0 y 7 incluidas ambas. El segundo como en el caso anterior, es también obligatorio, y debe de ser un número comprendido entre 1 y 4, que nos indica el número de banco que va a ocupar la ventana. Por último, el tercero es opcional, y nos permite ubicar en la memoria la ventana dentro del banco; esta dirección debe estar comprendida entre la 0 y la 16383.

Para más información y utilizar lo mejor posible los bancos de memoria, les informamos que cada ventana en modo 1 de $10^4 \times 10$ tiene una longitud de 1600 bytes con lo cual calzada al principio de un banco (en la dirección 0), nos quedaría libre hasta la 16383 para meter otras ventanas, datos o matrices dentro de un mismo banco de memoria. Con esto tenemos que la primera ventana podría situarse en la dirección 0 y la última en la dirección 14783, que sumado a los 1600 bytes no sobrepasan la dirección 16383.

Ahora vamos a ver con el segundo programa la velocidad que tiene de intercambio entre la memoria principal y ampliación. A la vez que comprobamos las posibilidades de estos comandos y su potencia.





Antes de seguir sería conveniente que observemos en el anterior programa la rapidez con que cambia la pantalla, dando la sensación de borroso. Sin embargo el programa al pulsar una tecla por primera vez crea una ventana, o continuación salva el banco n.º 1 la pantalla original para luego recuperarla de nuevo. Una vez creada y guardada en el banco la borra y espera que pulsamos cualquier tecla por segunda vez. Mientras tanto escribe un mensaje en pantalla que se repite sin cesar hasta que se pulsa una tecla. En ese momento carga la pantalla inicial y espera que se pulse una tecla.

Si ha leído todo lo anterior sobre la longitud de las ventanas se hará dos preguntas, ¿es posible crear ventanas en otros modos y otros tamaños? ¿cuánto ocupará en este caso cada ventana? Pues bien, sí es posible crear ventanas en otros tamaños y otros modos, y para facilitar la labor de programación acompañamos a continuación las siguientes fórmulas:

MODO 0 TAMAÑO = $(X_2 - X_1 + 1) * 4 * (Y_2 - Y_1 + 1) * 8$

MODO 1 O TAMAÑO = $(X_2 - X_1 + 1) * 2 * (Y_2 - Y_1 + 1) * 8$

MODO 2 O TAMAÑO = $(X_2 - X_1 + 1) * (Y_2 - Y_1 + 1) * 8$

Donde:

X1 es la coordenada del eje x de la izquierda de la ventana.

X2 es la coordenada del eje x de la derecha de la ventana.

Y1 es la coordenada del eje y de la parte superior de la ventana.

Y2 es la coordenada del eje y de la parte inferior de la ventana.

Si no asignamos bien el valor, por ejemplo más grande del admitido por el banco en el que estamos trabajando, el ordenador nos daría error y podría causar unos efectos extraños en la pantalla.

La tercera división se puede considerar la más importante, pues trata de las Matrices, Variables y Cadenas alfanuméricas. De nuevo son dos, las instrucciones que nos aportan a nuestro Basic, con sus respectivos parámetros opcionales, para cargar y salvar: Load y Save respectivamente. Su estructura sería:

Load, [Banco], [Localización], [Longitud], [Dirección del Banco]

Save, [Banco], [Localización], [Longitud], [Dirección del Banco]

El primer parámetro es el banco que vamos a utilizar; el segundo es la dirección de memoria en la cual comienzan los datos; el tercero la longitud de los datos; y por último, el cuarto es opcional, en él introduciremos la dirección del banco en la que se va a encontrar dichos datos, para poder alojar otros nuevos o distintos informaciones como puede ser alguna pantalla.

Ahora bien, nos encontramos con la necesidad de conocer la localización de nuestro matriz así como la longitud y dirección del banco. La localización es fácil, tan sólo debemos una vez dimensionada la matriz, solicitar dónde está localizado su primer elemento. Esto se consigue con:

```
10 DIM A(50)  
20 PRINT @ A(0)
```

En el caso de estar multidimensionado, tan sólo tendremos que ampliar el valor de A(0):

```
10 DIM A(5, 5, 16, 50)  
20 PRINT @ A(0, 0, 0, 0)
```

Con esto tenemos solucionado el primer problema que se nos presentaba. El segundo viene de la longitud de la matriz. En este caso hay que tener en cuenta de qué tipo son sus valores: si son números reales tendrá cinco bytes por elemento, si son enteros contará dos bytes por elemento y si son cadenas la longitud de bytes será variable.

Por ejemplo si la matriz es de:

```
10 DIM A(50)
```

en este caso serían 51 elementos. El número de Bytes es de $51 * 5 = 255$ Bytes.

```
10 DIM A(33, 27)
```

en este otro serían $33 * 27 = 891$ elementos. El número de bytes es de $891 * 5 = 4455$ Bytes.

Vamos a utilizar una matriz «A(50)» la cual tiene un total de 51 elementos y 255 bytes. Estos se encuentran en la dirección de PRINT @ A(0) y tienen una longitud de 255 bytes. El primero estará en la 0 el siguiente en la 255 el siguiente en la 510, etc.

En el tercer programa vamos a crear un fichero que nos lo rellene de valores para luego utilizarlos:



Ahora una vez creados los datos, borraremos este programa y crearemos el siguiente:

Este programa lo podemos tener en cinta o en disco, para luego utilizarlo en los bancos de memoria. Si queremos operar un poco con él, tendríamos que teclear este otro con el que podríamos sacar una idea particular y darle nuestra propia utilidad en futuros programas basado en la ampliación de memoria.

El cuarto grupo de instrucciones es la dedicada al manejo de almacenar las pantallas y ventanas, con la que se consigue un efecto de animación. Esto se consigue mediante tres nuevos comandos: High Low y Swap. Las dos primeras instrucciones no utilizan parámetros opcionales, colocándose independientemente en medio de nuestro programa. La tercera no lleva parámetros pero la podemos encontrar como parámetro dentro de las anteriores instrucciones. En caso de ir independiente, nos actúa como un intercambio de pantallas o datos entre dos zonas de memoria. Si va asociada la encontraremos como un valor que puede ser 0 ó 1, si queremos activarlo o no. Si no queremos ponerlo se toma por defecto el valor de 0, el cual es desactivado. Este lo podemos encontrar en las instrucciones siguientes:

```
Loads, [Banco], [Swap]  
Saves, [Banco], [Swap]  
Loadw, [Número Ventana], [Banco], [Dirección del Banco], [Swap]  
Savew, [Número Ventana], [Banco], [Dirección del Banco], [Swap]
```

Los comandos de Low y High nos dan la parte baja y alta de la pantalla respectivamente. Que podemos intercambiar sus valores con la tercera instrucción (Swap).

Bien ahora si modificamos el programa número 1, e introducimos las siguientes líneas:

```
5 MEMORY 16383: HIGH: REM o LOW modificar la memoria 135 IF  
SCREEN2=SCREEN/2 THEN T=TIME: WHILE  
TIME+20: WEND  
140 LOADS, SCREEN, 1: SWAP
```

Podremos ver el efecto que produce en la pantalla, utilizando el High o el Low en el programa número 1.

La última sección de comandos, serían tal vez los comandos destinados a la programación directa de bancos o modificación de éstos dentro de nuestros programas o los comerciales.



Este grupo lo podemos subdividir en dos grupos. El primero estorío formados por los comandos: Bank y Askrom. Estos dos comandos nos dan la posibilidad de trabajar en un banco determinado y hallar la información necesaria para trabajar en éste. La orden Askrom se acompaña de dos parámetros:

ASKRAM, [Opción], [Variable].

Dentro de la opción podemos elegir entre tres valores:

1- Nos da información sobre la cantidad de RAM utilizada.

2- Nos indica el número de bancos ocupados.

3- Asigna el valor 0 ó 1 dependiendo del estado de la RAM.

El segundo subgrupo está formado por otras dos instrucciones que son: Poke y Peek. Este nos modifica el valor de uno determinado función o parámetro de un banco. Los comandos Peek y Poke vienen acompañados de tres parámetros:

PEEK, [Banco], [Dirección del Banco], [Variable].

POKE, [Banco], [Dirección del Banco], [Variable].

Donde a la Variable se le asigna el Valor de la dirección del banco (de 0 o 16383).

Bajo la lupa

La ampliación recurre para obtener los 64 K extras de memoria RAM o un grupo de ocho integrados. Además incorpora cinco integrados que manejan las direcciones de memorias, y un circuito lógico PAL que controla los diferentes bancos de memoria.

El primer grupo de ocho chips son de 64 K cada uno de memoria dinámica, que hacen en total 512 K falsos. Esto es debido a que por el diseño de la postilla se precisa del total mencionado para obtener la memoria deseada.

De los cinco integrados del segundo grupo, dos de ellos se encargan de formar los cuatro bloques de 16 K que forman la memoria, mientras que los otros tres, tienen como misión el direccionamiento de éstos.

Problemas de compatibilidad

Como yo se ha comentado en la introducción, el periférico de DK' TRONICS está únicamente dirigido a los usuarios de los ordenadores CPC 464 y 664, lo cual excluye a lo voriente español del 464 denominado 472 que ha sido comercializada recientemente. Esto se debe a que esta última versión incorpora uno nuevo ROM poco amiga de la ampliación de memoria como hemos podido comprobar en nuestro banco de pruebas.

Francisco G.R.

PROGRAMAS

PROGRAMA 1

```

10 MODE 1
20 BORDER 0
30 FOR col=0 TO 3
40 INK col,0
50 NEXT col
60 FOR col=0 TO 3
70 PAPER col:CIS
80 ISAVES,col+1
90 NEXT col
100 TINI 0,1:INI 1,6:INI 2,2:INI 3,
13
110 PEN 1:PAPER 0
120 WHILE INKEY$=""
130 END pantalla=1 TO 4
140 ILOADS,pantalla
150 NEXT pantalla
160 WEND
170 END

```

PROGRAMA 2

```

10 MODE 1
20 FOR i=1 TO 10
30 PRINT STRING$(39,CHR$(217));
40 PRINT STRING$(39,CHR$(218));
50 NEXT i
60 CALL #BB06:REM Espera pulsar una tecla
70 WINDOW #1,INT(RND(1)*1911),INT(RND(0)*19+INT(RND(1)*5117)),INT(RND(1)*14+1),INT(RND(0)*14+INT(RND(1)*0+5))
80 PEN #1,2:PAPER #1,3
90 ISAVES,1,1:REM Salva el contenido de la pantalla en la RAM
100 CLS #1:REM Borra la pantalla
110 WHILE INKEY$="":REM Espera pulsar una tecla por segunda vez
120 PRINT #1,"VENTANA MILROHOBBY"
130 WEND
140 ILOADW,1,1:RFM Restablece la pantalla
150 GOTO 60

```

PROGRAMA 3

```

10 OPENOUT "stock.dat"
20 FOR week=1 TO 52
30 FOR item=1 TO 60
40 PRINT #9,INT(RND(1)*3000+100)
50 NEXT item
60 NEXT week
70 CLOSEOUT
80 END

```

PROGRAMA 4

```

10 DIM stock(60)
20 INPUT "Leemos ficheros (S/N)":ans$
30 IF LOWER$(ans$) = "S" OR LOWER$(ans$) = "SI" THEN GOSUB 1000
1000 REM subrutina para leer datos del disco
1010 OPENIN "stock.dat"
1020 FOR week=1 TO 52
1030 FOR item=1 TO 60
1040 INPUT #9,stock(item)

```

```

1050 NEXT item
1060 ISAVED,4,0,stcl(0),61$5,wrhi*05
1070 NEXT week
1080 CLOSERIN
1090 RETURN

```

PROGRAMA 5

```

10 DIM stock(60)
20 INPUT "Leemos fichero (S/N)":ans$
30 IF LOWER$(ans$) = "S" OR LOWER$(ans$) = "SI" THEN GOSUB 1000
40 RFM
100 MODE 2
110 LOCATE 1,1
120 INPUT "Cita anidamos":tempo
130 IF tempo = 1 OR tempo = 60 THEN 120
140 CLS:LOCATE 20,1
150 PRINT "Borra para cada item":item
160 LOCATE 10,25
170 PRINT "Ene Feb Mar Abr
May Jun Jul Ago Sep Oct
Nov Dic":RFM 3 espacios entre e
110s
180 FOR loop=1 TO 4
190 LOCATE 1,24:loop$:
200 PRINT $1$":loop$:"000"
210 NEXT loop
220 MOVE 60,768:DRAW 60,0:DRAW 61,0
:DRAW 61,368:MOVE 640,24:DRAW 411,24
230 FOR loop=1 TO 4
240 MOVL 48,loop*80+24:DRAW 60,1:loop$:
250 NEXT loop
260 FOR week=1 TO 52
270 IF week/2 = week \2 THEN n=1 ELSE
n=2
280 ILOADD,4,0,stcl(0),61$5,week*05
290 ycoord=stock(item)/4000*320
AND 4092
300 FOR xcoord=1 TO 11 STEP n
310 MOVE 49+xcoord+week*11,ycoord+2
6:DRAW 49+coor+d+week*11,26
320 NEXT xcoord
330 NEXT week
340 GOTO 110
1000 REM subrutina de leer datos de
1 disco
1010 OPENIN "stock.dat"
1020 FOR week=1 TO 52
1030 FOR item=1 TO 60
1040 INPUT #9,stock(item)
1050 NEXT item
1060 ISAVI D,4,0,stcl(0),61$5,wrhi*05
1070 NLXI week
1080 CLOSERIN
1090 RETURN

```

AMPLIACION DE MEMORIA

Distribuidor: Microware.
Precio: 15.350 ptas.
C/ Clara del Rey, 58. Madrid.

AMSTRAD CPC - 464

AMSTRAD



ORDENADOR

SERIE CPC

UNIDAD CENTRAL. MEMORIAS

- Microprocesador Z80A - 64K RAM ampliables - 32K ROM ampliables
- TECLADO • Teclado profesional con 24 teclas en 3 bloques - Hasta 32 teclas programables - Teclado redefinible
- PANTALLA • Monitor RGB verde (12") o color (14")

	Normal	Alta Res.	Multicolor
Col x líneas	40 x 25	80 x 25	20 x 25
Colores	4 col.	16 col.	16 de 27
Puntos	320 x 200	640 x 200	160 x 2

- Se pueden definir hasta 8 ventanas de texto y 1 de gráficos
- SONIDO
 - 3 canales de 8 octavas moduladas independientemente - Altavoz interno regulable - Salida estéreo
 - BASIC
 - Locomotive BASIC ampliado en ROM - Incluye los comandos AFTER y EVERY para control de interrupciones

AMSTRAD CPC 464

- Cassette incorporada con velocidad de grabación (1 ó 2 Kbaudios) controlada desde Basic
- CONECTORES
 - Bus PCP multiuso, Unidad de Disco exterior, paralelo Centronics, salida estéreo, joystick, lápiz óptico, etc.
- SUMINISTRO • Ordenador con monitor variante o color - 8 cassettes con programas - Libro "Guía de Referencia BASIC para el programador" - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA

TODO POR 59.900 Pts. (monitor verde)
90.900 Pts. (monitor color)

AMSTRAD CPC 6128

- UNIDAD DE DISCO • Unidad incorporada para disco de 3" con 180K por cara
- SISTEMAS OPERATIVOS
 - AMSDOS, CP/M 2.2, CP/M Plus (3.0)
- CONECTORES • Bus PCP multiuso, paralelo Centronics, cassette exterior, 2.ª Unidad de Disco, salida estéreo, joysticks, lápiz óptico, etc.
- SUMINISTRO • Ordenador con monitor variante o color - Disco con CP/M 2.2 y lenguaje DR LOGO - Disco con CP/M Plus y utilidades - Disco con 6 programas de obsequio - Manual en castellano - Garantía Oficial AMSTRAD ESPAÑA

TODO POR 84.900 Pts. (monitor verde)
119.900 Pts. (monitor color)

PCW-8256

AMSTRAD CPC-6128



ES AMSTRAD *¡Increíble!*

AMSTRAD PCW 8256

UNIDAD CENTRAL MEMORIAS

- Microprocesador Z80A - 256K RAM de las que 112K se utilizan como disco RAM
- **TECLADO** • Teclado profesional en castellano (ñ, acento...) de 82 teclas
- **PANTALLA** • Monitor verde de alta resolución - 90 columnas x 32 líneas de texto
- **UNIDAD DE DISCO** • Disco de 3" y 173K por cara - Opcionalmente, 2.ª Unidad de Disco de 1 Mbyte integrable
- **SISTEMA OPERATIVO** • CP/M Plus de Digital Research • **IMPRESORA** • Alta calidad (NLQ) a 20 c.p.s. - Calidad estándar a 30 c.p.s. - Papel continuo u hojas sueltas - Alineación automática del papel - Caracteres normales, comprimidos, expandidos, control del peso de letra (normal, cursiva, negrita, subíndices, superíndices, subrayado, etc.)
- **Opciones** • Kit de Ampliación a 128K RAM y 2.ª Unidad de Disco - Interface Serie RS 232C y paralelo

Centronics • **SUMINISTRO** • Ordenador completo con teclado, pantalla, Unidad de Disco e Impresora - Discos con el procesador de Texto Laser Script, CP/M Plus, Mallard, BASIC, DR. LOGO y diversas utilidades - Manuales en castellano - Garantía Oficial AMSTRAD ESPAÑA

TODO POR 129.900 Pts.



Los más prestigiosos paquetes de Software Profesional, en formato AMSTRAD a "precios AMSTRAD"

Existe también la versión **PCW 8512** con 512K RAM y la 2.ª Unidad de Disco de 1 Mbyte incorporada **PVP. 174.900 Pts.**

* El **PCW 8256** puede de utilizarse como terminal y en comunicaciones.

El I.V.A. no está incluido en los precios.

NOTA: Es muy importante verificar la garantía del aparato ya que sólo **AMSTRAD ESPAÑA** puede garantizarse la ordenada reparación y sobre todo materiales de repuesto oficiales (Monitor, ordenador, cassette o unidades de discos).

AMSTRAD

ESPAÑA

Avenida del Mediterráneo, 9. Tel. 433 45 43 - 433 18 76.
28007 MADRID

Delegación Cataluña: Tarragona, 110 - Tel. 325 10 56.
08015 BARCELONA

AMSTOCK-W

Realizado por Rafael Sánchez de López

El PCW8256, por su precio, configuración y prestaciones, viene que ni pintado para realizar aplicaciones de gestión, y en este sentido existen cada vez más cantidad de programas comerciales que pretenden resolver los posibles problemas con que un usuario empresarial puede enfrentarse.

Una de las más comunes es el control de stocks; nosotros presentamos un completo programa que resuelve esto, escrito íntegramente en Mallard Basic, con salida impresa incluida, y, además, gratis. Creemos que merece la pena teclearlo.



Controlar un almacén es una de las tareas más lentas, pero a la vez más importante de las que se llevan a cabo en cualquier actividad. Su buen funcionamiento dará buenos datos para elaborar costos, sobre todo en actividades de fabricación. A los almacenistas les ofrecerá siempre una visión actualizada de sus existencias.

Del almacén debemos conocer en todo momento la situación de los elementos que lo componen, saber qué materiales tenemos, cuánta cantidad hay de los mismos, quién los ha servido, el precio al que los hemos comprado, las existencias mínimas para poder trabajar mientras ese material es repuesto por el suministrador y el valor total de cada material.

En el momento de la entrada de material en almacén, se debe contabilizar dicha entrada, haciendo los aumentos necesarios de cantidad de material, como de precio del mismo, para salidas el mismo método pero disminuyendo estos datos de la ficha del artículo.

En cuanto a valoración del almacén existen tres métodos fundamentales, nosotros hemos empleado el LIFO, es decir, que el material del almacén siempre se actualiza al último precio indicado.

Mediante el presente programa, se evita la manipulación manual de fichas así como la búsqueda, variación y archivado de las fichas de material. La ventaja fundamental es que



todo nuestro archivo se encuentra en un solo disco y que en él, se efectúan todos las operaciones clásicas del manejo de almacén. Con nuestro programa, tenemos en todo momento actualizada, toda la información, así como la posibilidad de inventariar en el acto, conociendo el valor de nuestro almacén puntualmente. Podremos controlar todos estos elementos de una manera rápida y sencilla; en todo momento y a través de la consola tendremos la información para poder acceder a las distintas opciones, y siempre se nos indica lo que debemos hacer para llevar a cabo nuestros propósitos.

ACceso A MENU Y SUBMENUS

El acceso a las opciones, está basada en sencillas menús mediante los cuales nos podremos mover a través del mapa del programa. Tenemos en todo momento: la libertad de descender a un submenu, activar una opción o regresar al menú principal.

El fichero del almacén está formado por fichas individuales de cada artículo y en las cuales tenemos archivadas, la clave del artículo, su denominación, las existencias, las existencias mínimas, el proveedor que nos lo ha suministrado, el precio unitario y el valor total de cada artículo.

Una vez cargado el programa nos aparecerá en la pantalla el menú principal, éste consta de cinco opciones a las cuales podemos dirigirnos con sólo pulsar el número que antecede a cada opción.

La opción primera es el acceso a los ficheros, es decir, a la creación, aumento o borrado de las fichas de nuestro almacén.

La segunda opción nos traslada a la parte más importante del programa ya que con ésta será con la que más trabajemos pues tenemos en nuestras manos las entradas y salidas del material, las consultas de las fichas y las modificaciones de las mismas.

Con la opción tres pasamos a poder listar tanto en pantalla como en impresora las fichas de materiales así como las fichas del material que se encuentre bajo mínimos.

Con la opción cuatro tenemos la posibilidad de inventariar el almacén. Tendremos la opción de la pantalla o la impresora.

La última opción es la más obvia ya que únicamente sirve para abandonar el programa.

Todas estas opciones del menú principal conducen a unos submenús de los cuales no hace falta explicar nada ya que directamente conducen a operaciones con las fichas.

En todo momento se nos indicará en la pantalla del monitor y en la parte posterior dónde nos encontramos y en la parte inferior se nos dice qué debemos hacer para seguir adelante. Todos los submenús tienen una última opción que es la de volver al menú principal.

AMSTOCK-W Y LA IMPRESORA

Cuando entremos en el capítulo de los listados hay que tener en cuenta que el programa está preparado para trabajar con hojas sueltas, es decir, con papel DIN-A4, por tanto antes de acceder al listado hay que alimentar la impresora.

Para alimentar la impresora basta con centrar el papel y dejarle que descance sobre el rodillo, una vez hecho esto se desplaza hacia delante la palanca que se encuentra al lado del mando del rodillo. El papel se introducirá sola, en este momento regresamos la palanca a su posición y pulsamos en el zócalo del ordenador la tecla «SAL». Una vez alimentada la impresora escogeremos la opción

Para... PCW

de listar que nos interese y ésta se ejecutará expulsando al terminar la hoja aunque ésta no hubiera sido impresa en su totalidad.

A la hora de listar el material se ha optado por disminuir el tipo de letra con el objeto de poder disponer en un solo listado de todos los elementos que componen el almacén.

ULTIMOS CONSEJOS

Cuando terminemos el trabajo es recomendable inventariar el almacén ya que a la vez de evitarnos disgustos de fallos en los discos nos proporcionará tanto a los operadores del programa como a los distintos departamentos de la empresa los datos de situación del almacén. Hay que hacer notar que con los datos del almacén trabajan los departamentos de Contabilidad, Comercial, Administración, Producción y Gerencia.

DEFINICION DE VARIABLES

ESC\$	Código de control para escapes.
SON\$	Produce el pitido.
CLR\$	Barrado de pantalla y cursor al origen.
CLAVE\$	Clave del artículo para definir el campo.
DENOMINACION\$	Nombre del artículo para definir el campo.
PROVEEDOR\$	Nombre del proveedor para definir el campo.
CANTIDAD\$	Unidades del artículo para definir el campo.
EXMIN\$	Existencias mínimas para definir el campo.
PRECUNI\$	Precio unitario del artículo para definir el campo.
PRECTO\$	Precio total para definir el campo.
OPC\$	Evaluá la tecla pulsada en los menús.
DEN\$	Nombre del artículo.
PRO\$	Nombre del proveedor.
UDS	Unidades.
EX.MIN	Existencias mínimas.
PT.UNI	Precio unitario.
CLAVE	Clave del artículo.
X Y A B	VARIABLES PARA CONTADORES.
CANTIDADE	Unidades de entrada de material.
PRECUNIE	Precio unitario de entrada de material.
CANTIDADS	Unidades de salida de material.
CLAVEN	Nueva clave en modificaciones.
DENOMINACIONN\$	Nueva denominación en modificaciones.
PROVEEDORN\$	Nuevo proveedor en modificaciones.
EXISMNN	Nuevas existencias mínimas en modificaciones.

ESTRUCTURA

Normalmente nos será imposible recordar todos los códigos o claves de los materiales del almacén, por lo que recomendamos antes de empezar a trabajar sacar un listado de materiales con lo cual tendremos impresos estos datos. Si sabemos la clave pero tenemos dudas podemos evitarnos el listar ya que podemos consultar con la opción tres del segundo submenú correspondiente al mantenimiento de fichero del menú principal.

Una última advertencia para el manejo del programa, como hemos indicado anteriormente, nunca os encontraréis solos en una pantalla sin saber qué hacer ya que siempre se da una opción para seguir el programa. Limitaros por tanto a seguir estas instrucciones.

En cuanto a la estructuración del programa ya hemos indicado que se basa en menús, por lo tanto el seguimiento del mismo es fácil gracias a las sentencias ON GOTO, que nos indican siempre a dónde nos dirigimos.

Cuando nos dispongamos a trabajar con este programa hay que tener en cuenta las características propias del ordenador, que como sabemos dispone de siete lenguajes. Cuando el ordenador se pone en funcionamiento se adopta implícitamente el siete que es el español pero tiene la desventaja de perder ciertos caracteres del ASCII básico, por tanto se recomienda reinicializar el ordenador, cargar del CPM, cuando nos pida la primera sentencia le contestamos Language 0.

Y seguidamente le indicamos que cargue el Basic. Hecho esto ya podemos empezar a copiar o introducir el disco con el programa.



10-80	Preparación de variables de control y definición de funciones.
100-210	Presentación menú principal.
230-370	Presentación primer submenú.
380-600	Creación inicial de fichero.
610-880	Aumentar fichas del fichero.
890-980	Borrar fichas.
990-1140	Presentación del segundo submenú.
1150-1300	Presentación del tercer submenú.
1310-1500	Entrada de material.
1510-1710	Salida de material.
1720-2030	Consultas de fichas.
2040-2170	Presentación de submenú para modificaciones.
2180-2320	Modificación de la clave.
2330-2480	Modificación y denominación del artículo.
2490-2640	Modificación nombre de proveedor.
2650-2800	Modificación de las existencias mínimas.
2810-3130	Listado de material por pantalla.
3140-3280	Listado de mínimos por pantalla.
3290-3590	Listado de material por impresora.
3600-3750	Listado de mínimos por impresora.
3760-4140	Inventario.
4150-5140	Listado del inventariado por impresora.

```

10 REM *** Control de Stock ***
20 REM Rafael Sanchez de Lopez Feb
rero 1.986
30 REM *** preparacion de variables ***
40 esc$=CHR$(27): son$= CHR$(7): cl
r$= esc$+"E"+esc$+"H":
50 REM *** funciones definidas ***
60 DEF FNlocate$(x,y,texto$)=esc$+"Y"+CHR$(32+y)+CHR$(32+x)+textos
70 REM *** MENU PRINCIPAL ***
80 PRINT clr$
90 PRINT FNlocate$(38,3,"MENU PRINC
IPAL"):PRINT FNLOCATE$(38,4,"-----")
100 PRINT FNlocate$(30,8,"*****")
110 PRINT FNlocate$(30,9,"* 1 - acc
eso a ficheros ")
120 PRINT FNlocate$(30,10,"* 2 - ma
ntenimiento ficheros ")
130 PRINT FNlocate$(30,11,"* 3 - li
stados ")
140 PRINT FNlocate$(30,12,"* 4 - in
ventariado ")
150 PRINT FNlocate$(30,13,"* 5 - ab
andonar programa ")
160 PRINT FNlocate$(30,14,"*****")
170 PRINT FNlocate$(50,27,"pulse la
opcion deseada (1-5)")
180 opc$=INKEY$: IF opc$="" THEN 18
0
190 IF VAL(opc$)<1 OR VAL(opc$)>5 T
HEN PRINT FNlocate$(50,27,son$) ELSE
210
200 GOTO 180
210 ON VAL(opc$) GOTO 230,990,1150,
3760,220
220 END
230 REM *** acceso a ficheros ***
240 clave$="1234":PRINT clr$
250 PRINT TAB(60)"opcion 1 del menu
principal"
260 PRINT FNlocate$(38,5,"FICHEROS")
: PRINT FNlocates$(38,6,"-----")
270 PRINT FNlocate$(30,8,"*****")
280 PRINT FNlocate$(30,9,"* 1 - cre
acion inicial ")
290 PRINT FNlocate$(30,10,"* 2 - au
mentar fichas ")
300 PRINT FNlocate$(30,11,"* 3 - bo
rrar fichas ")
310 PRINT FNlocate$(30,12,"* 4 - vo
lver menu principal ")
320 PRINT FNlocate$(30,13,"*****")
330 PRINT FNlocate$(50,27,"pulse la
opcion deseada (1-4)")
340 opc$=INKEY$: IF opc$="" THEN 34
0
350 IF VAL(opc$)<1 OR VAL(opc$)>4 T
HEN PRINT FNlocate$(50,27,son$) ELSE
370
380 GOTO 340
390 ON VAL(opc$) GOTO 380,610,890,8
0
380 REM *** creacion inicial ***
390 clave=
400 PRINT clr$
410 PRINT TAB(60)"opcion 1 del subm
enu 1"
420 INPUT"denominacion ", den$
430 IF den$="" THEN 230
440 INPUT"proveedor ",pro$
450 INPUT "cantidad ",uds
460 INPUT"existencias minimas ",ex
_min
470 INPUT "precio unitario ",pt.uni
480 OPEN"R",1,"base"
490 FIELD 1,2 AS clave$,20 AS denom
inaciones$,20 AS proveedor$, 2 AS can
tidades$,2 AS eximin$,4 AS precuni$,4
AS precio$
500 RSET clave$=MKI$(clave)
510 LSET denominacion$=den$
520 LSET proveedor$=pro$
530 RSET cantidad$=MKI$(uds)
540 RSET eximin$=MKI$(ex_min)

```

Unicamente hemos observado con este método una diferencia y es la de que al listar por impresora el código Barrado (I) es sustituido por el símbolo de pesetas. Para generarla basta pulsar EXTRA + 4.

Las opciones más delicadas del programa son aquéllas con las que destruimos información como son la opción de borrar fichas o la de modificarlas, el programa abedece la instrucción pero no la confirma, por tanto tener cuidado y asegurar bien los datos cuando manejéis estas opciones.

Una curiosidad es la de observar cómo se actualizan las fichas, podéis hacer una prueba con las opciones de entrada o salida de material y luego con la instrucción de consulta llámala a la ficha y observáis cómo las cantidades, existencias y precios se han modificado de acuerdo con las operaciones que habéis efectuado.

```

550 RSET precuni$=MKS$(pt,uni)
560 RSET precio$=MKS$(pt,uni*uds)
570 PUT 1,clave
580 PUT 1,999
590 CLOSE
600 clave=clave+1;GOTO 400
610 REM *** aumentar fichas ***
620 PRINT clr$
630 OPEN"R",1,"base"
640 FIELD 1,2 AS clave$,20 AS denominacion$,20 AS proveedor$, 2 AS cantidad$,2 AS eximin$,4 AS precuni$,4 AS precio$
650 GET 1,999
660 clave=CVI(clave$);clave=clave+1
670 CLOSE
680 PRINT clr$
690 PRINT TAB(60)"opcion 2 del submenu 1"
700 INPUT"denominacion ", den$
710 IF den$="" THEN 200
720 INPUT"proveedor ",pros$
730 INPUT "cantidad ",uds
740 INPUT"existencias minimas ",ex,min
750 INPUT "precio unitario ",pt,uni
760 OPEN"R",1,"base"
770 FIELD 1,2 AS clave$,20 AS denominacion$,20 AS proveedor$, 2 AS cantidad$,2 AS eximin$,4 AS precuni$,4 AS precio$
780 RSET clave$=MKI$(clave)
790 LSET denominacion$=den$
800 LSET proveedor$=pros$
810 RSET cantidad$=MKI$(uds)
820 RSET eximin$=MKI$(ex,min)
830 RSET precuni$=MKS$(pt,uni)
840 RSET precio$=MKS$(pt,uni*uds)
850 PUT 1,clave
860 PUT 1,999
870 CLOSE
880 clave=clave+1;GOTO 680
890 REM *** borrar fichas ***
900 PRINT clr$;PRINT TAB(60)"opcion 3 del submenu 1"
910 PRINT FNlocate$(25,12,"clave de la ficha ");INPUT clave
920 OPEN"R",1,"base"
930 FIELD 1,2 AS clave$,20 AS denominacion$,20 AS proveedor$, 2 AS cantidad$,2 AS eximin$,4 AS precuni$,4 AS precio$
940 GET 1,clave
950 LSET denominacion$="";LSET proveedor$="";RSET cantidad$=MKI$(0);RSET eximin$=MKI$(0);RSET precuni$=MKS$(0);RSET precio$=MKS$(0)
960 PUT 1,clave
970 CLOSE
980 GOTO 230
990 REM *** mantenimiento ficheros ***
1000 PRINT clr$
1010 PRINT TAB(60)"opcion 2 del menu principal"
1020 PRINT FNlocate$(33,5,"MANTELLO MIENTO FICHEROS"); PRINT FNlocate$(33,6,"-----")
1030 PRINT FNlocate$(30,8,"*****")
1040 PRINT FNlocate$(20,9,"* 1 - entrada material ")
1050 PRINT FNlocate$(30,10,"* 2 - salida material ")
1060 PRINT FNlocate$(30,11,"* 3 - consultas ")
1070 PRINT FNlocate$(30,12,"* 4 - modificaciones ")
1080 PRINT FNlocate$(30,13,"* 5 - volver menu principal ")
1090 PRINT FNlocate$(30,14,"*****")
1100 PRINT FNlocate$(50,27,"pulse 1 a opcion deseada (1-5)")
1110 opc$=INKEY$; IF opc$="" THEN 1
110
1120 IF VAL(opc$)<1 OR VAL(opc$)>5 THEN PRINT FNlocate$(50,27,son$) ELSE 1140
1120 GOTO 1110
1140 ON VAL(opc$) GOTO 1310,1510,17
20,2040,80
1150 REM *** listados ***
1160 PRINT clr$
1170 PRINT TAB(60)"opcion 3 del menu principal"
1180 PRINT FNlocate$(38,5,"LISTADOOS ");PRINT FNlocate$(38,6,"-----")
1190 PRINT FNlocate$(25,8,"*****")
1200 PRINT FNlocate$(25,9,"* 1 - listado material pantalla ")
1210 PRINT FNlocate$(25,10,"* 2 - 1 listado minimos pantalla ")
1220 PRINT FNlocate$(25,11,"* 3 - 1 listado material impresora ")
1230 PRINT FNlocate$(25,12,"* 4 - 1 listado minimos impresora ")
1240 PRINT FNlocate$(25,13,"* 5 - volver menu principal ")
1250 PRINT FNlocate$(25,14,"*****")
1260 PRINT FNlocate$(50,27,"pulse 1 a opcion deseada (1-5)")
1270 opc$=INKEY$; IF opc$="" THEN 1
270
1280 IF VAL(opc$)<1 OR VAL(opc$)>5 THEN PRINT FNlocate$(50,27,son$) ELSE 1300
1290 GOTO 1270
1300 ON VAL(opc$) GOTO 2810,3140,32
90,3600,80
1310 REM *** entrada de material ***
1320 PRINT clr$
1330 PRINT FNlocate$(25,12,"clave del material : ");INPUT clave
1340 OPEN"R",1,"base"
1350 FIELD 1,2 AS clave$,20 AS denominacion$,20 AS proveedor$, 2 AS cantidad$,2 AS eximin$,4 AS precuni$,4 AS precio$
1360 GET 1, clave
1370 PRINT clr$
1380 PRINT FNlocate$(25,10,"cantidad del material : ");INPUT cantidad
1390 PRINT FNlocate$(25,11,"precio unitario : ");INPUT precunie
1400 cantidad=CVI(cantidad$);RSET cantidad$=MKI$(cantidad$+cantidad)
1410 RSET precuni$=MKS$(precunie)
1420 precio$=CVS(precio$);RSET precio$=MKS$(precunie*(cantidad$+cantidad))
1430 PUT 1,clave
1440 CLOSE
1450 PRINT clr$
1460 PRINT FNlocate$(15,10,"el material ha sido contabilizado y archivado la ficha correspondiente")
1470 PRINT FNlocate$(50,27,"hay mas entradas de material (s/n)")
1480 a$=INKEY$; IF a$="" THEN 1480
1490 IF a$="s" OR a$="S" THEN 1310 ELSE 990
1510 REM *** salida de material ***
1520 PRINT clr$
1530 PRINT FNlocate$(25,12,"clave del material : ");INPUT clave
1540 OPEN"R",1,"base"
1550 FIELD 1,2 AS clave$,20 AS denominacion$,20 AS proveedor$, 2 AS cantidad$,2 AS eximin$,4 AS precuni$,4 AS precio$
1560 GET 1, clave
1570 PRINT clr$
1580 PRINT FNlocate$(25,10,"cantidad del material : ");INPUT cantidad
1590 cantidaddb=CVI(cantidad$);RSET cantidad$=MKI$(cantidad$+cantidad)
1610 REM RSET precuni$=MKI$(precuni$)
1620 REM precio$=CVI(precio$);RSET precio$=MKI$(precio$+precio$*cantidad)
1630 precunis=CVS(precuni$);RSET precuni$=MKS$(precunis);RSET precio$=MKS$(precunis*(cantidad$+cantidad))
1640 PUT 1,clave
1650 CLOSE
1660 PRINT clr$
1670 PRINT FNlocate$(15,10,"el material ha sido contabilizado y archivado la ficha correspondiente")
1680 PRINT FNlocate$(50,27,"hay mas entradas de material (s/n)")
1690 a$=INKEY$; IF a$="" THEN 1480
1710 IF a$="s" OR a$="S" THEN 1510 ELSE 990
1720 REM *** consulta de ficheros ***
1730 PRINT clr$
1740 PRINT FNlocate$(30,10,"numero de ficha a consultar");INPUT clave
1760 OPEN"R",1,"base"
1770 FIELD 1,2 AS clave$,20 AS denominacion$,20 AS proveedor$, 2 AS cantidad$,2 AS eximin$,4 AS precuni$,4 AS precio$
1780 GET 1,clave
1790 clave=CVI(clave$); cantidad=CVI(cantidad$); exi=CVI(eximin$); precuni=CVS(precuni$); precio=CVS(precio$)
1800 PRINT clr$
1810 CLOSE
1820 PRINT TAB(10)STRING$(74,"*")
1830 PRINT TAB(10) "*";STRING$(72,"*")
1840 PRINT TAB(10)* ficha numero
";clave;" *
1850 PRINT TAB(10)*";STRING$(72,"*")
1860 PRINT TAB(10)* denominacion ";denominacion";";proveedor";TAB(83)*"
1870 PRINT TAB(10)*";STRING$(72,"*")
1880 PRINT TAB(10)*";STRING$(70,"*")
1890 PRINT TAB(10)*";STRING$(72,"*")
1900 PRINT TAB(10)* unidades
precio unitario
precio total "
1910 PRINT TAB(10)*";STRING$(72,"*")
1920 PRINT TAB(10)*";STRING$(70,"*")
1930 PRINT TAB(10)*";STRING$(72,"*")
1940 PRINT TAB(10)*";STRING$(72,"*")
1950 PRINT TAB(10)*";TAB(15)cantidad;TAB(50)precuni;TAB(70)precio;TAB(83)*"
1960 PRINT TAB(10)*";STRING$(72,"*")
1970 PRINT TAB(10)*";STRING$(72,"*")
1980 PRINT TAB(10)*";TAB(55)existencias minimas;exi;TAB(83)*"
1990 PRINT TAB(10)*";STRING$(72,"*")
2000 PRINT TAB(10)STRING$(74,"*")
2010 PRINT FNlocate$(50,27,"pulse una tecla para seguir")
2020 a$=INKEY$;IF a$="" THEN 2020
2030 GOTO 1000
2040 REM *** modificaciones ***
2050 PRINT clr$
2060 PRINT TAB(60)"opcion 4 del submenu"
2070 PRINT FNlocate$(25,8,"*****")
2080 PRINT FNlocate$(25,9,"* 1 - modificacion clave ")
2090 PRINT FNlocate$(25,10,"* 2 - modificacion denominacion ")
2100 PRINT FNlocate$(25,11,"* 3 - modificacion proveedor ")
2110 PRINT FNlocate$(25,12,"* 4 - modificacion existencias minimas ")
2120 PRINT FNlocate$(25,13,"* 5 - v

```

```

olver submenu          *
)
2130 PRINT FNlocate$(25,14,"*****"
*****")
)
2140 PRINT FNlocate$(50,27,"pulse 1
a opcion deseada (1-5)")
2150 opc$=INKEY$; IF opc$="" THEN 2
150
2160 IF VAL(opc$)<1 OR VAL(opc$)>5
THEN PRINT FNlocate$(50,27,son$); G
OTO 2150
2170 ON VAL(opc$) GOTO 2180,2330,24
90,2650,990
2180 REM *** modificacion clave ***
2190 PRINT clr$
2200 PRINT FNlocate$(25,12,"clave d
el material ");;INPUT clave
2210 PRINT FNlocate$(25,13,"nueva c
lave ");;INPUT claven
2220 PRINT clr$
2230 OPEN"R",1,"base"
2240 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS prectos$,54
2250 GET 1,clave
2260 PUT 1,claven
2270 CLOSE
2280 PRINT clr$
2290 PRINT FNlocate$(15,10,"el mate
rial ha sido contabilizado y archiv
ado la ficha correspondiente")
2300 PRINT FNlocate$(50,27,"hay mas
entradas de material (s/n)")
2310 a$=INKEY$; IF a$="" THEN 2310
2320 IF a$="s" OR a$="S" THEN 2180
ELSE 2040
2330 REM *** modificacion denominac
ion ***
2340 PRINT clr$
2350 PRINT FNlocate$(25,12,"clave d
el material ");;INPUT clave
2360 PRINT FNlocate$(25,13,"nueva d
enominacion ");;INPUT denominacionn
$
2370 PRINT clr$
2380 OPEN"R",1,"base"
2390 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS prectos$,54
2400 GET 1,clave
2410 LSET denominacion$=denominacio
nn$
2420 PUT 1,clave
2430 CLOSE
2440 PRINT clr$
2450 PRINT FNlocate$(15,10,"el mate
rial ha sido contabilizado y archiv
ado la ficha correspondiente")
2460 PRINT FNlocate$(50,27,"hay mas
entradas de material (s/n)")
2470 a$=INKEY$; IF a$="" THEN 2470
2480 IF a$="s" OR a$="S" THEN 2330
ELSE 2040
2490 REM *** modificacion proveedor
 ***
2500 PRINT clr$
2510 PRINT FNlocate$(25,12,"clave d
el material ");;INPUT clave
2520 PRINT FNlocate$(25,13,"nuevo p
roveedor ");;INPUT proveedorn$
2530 PRINT clr$
2540 OPEN"R",1,"base"
2550 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS prectos$
2560 GET 1,clave
2570 LSET proveedor$=proveedorn$
2580 PUT 1,clave
2590 CLOSE
2600 PRINT clr$
2610 PRINT FNlocate$(15,10,"el mate
rial ha sido contabilizado y archiv
ado la ficha correspondiente")
2620 PRINT FNlocate$(50,27,"hay mas
entradas de material (s/n)")

2630 a$=INKEY$; IF a$="" THEN 2630
2640 IF a$="s" OR a$="S" THEN 2490
ELSE 2040
2650 REM *** modificacion existenci
as minimas ***
2660 PRINT clr$
2670 PRINT FNlocate$(25,12,"clave d
el material ");;INPUT clave
2680 PRINT FNlocate$(25,13,"nuevas
existencias minimas ");;INPUT eximi
nn
2690 PRINT clr$
2700 OPEN"R",1,"base"
2710 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS prectos$
2720 GET 1,clave
2730 RSET eximin$=MKI$(eximinn)
2740 PUT 1,clave
2750 CLOSE
2760 PRINT clr$
2770 PRINT FNlocate$(15,10,"el mate
rial ha sido contabilizado y archiv
ado la ficha correspondiente")
2780 PRINT FNlocate$(50,27,"hay mas
modificaciones (s/n)")
2790 a$=INKEY$; IF a$="" THEN 2630
2800 IF a$="s" OR a$="S" THEN 2650
ELSE 2040
2810 REM *** listado material por p
antalla ***
2820 PRINT clr$
2830 OPEN"R",1,"base"
2840 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS prectos$
2850 GET 1,999;clave=CVI(clave$);a=
22;b=1
2860 IF clave<a THEN 3020
2870 PRINT "clave";TAB(10)"denomin
acion";TAB(30)"proveedor";TAB(51)"c
antidad";TAB(61)"exi/min";TAB(70)"p
rec/uni";TAB(80)"prec/total"
2880 PRINT STRING$(90,"-")
2890 FOR y=b TO a
2900 clave=clave-1
2910 GET 1,y
2920 clave=CVI(clave$);PRINT USING
"##"; clave;
2930 PRINT TAB(8)denominacion$;TAB(
29)proveedor$;
2940 cantidad=CVI(cantidad$);PRINT
TAB(53) USING "##";cantidad;;exi
min=CVI(eximin$);PRINT TAB (61) US
ING "##";eximin;
2950 precuni=CVS(precuni$);PRINT TA
B(69) USING "##";precuni;;prec
to=CVS(preco$);PRINT TAB (79)USING
"##";preco
2960 NEXT Y
2970 PRINT FNlocate$(50,27,"pulse u
na tecla para continuar")
2980 opcion$=INKEY$;IF opcion$="" T
HEN 2980
2990 IF clave=0 THEN 3010
3000 PRINT clr$;b=b+22;IF clave<22
THEN a=clave+22 :GOTO 2870
3010 CLOSE;GOTO 1150
3020 PRINT "clave";TAB (10)"denomin
acion";TAB(30)"proveedor";TAB(51)"c
antidad";TAB(61)"exi/min";TAB(70)"p
rec/uni";TAB(80)"prec/total"
3021 PRINT STRING$(90,"-")
3022 FOR x=1 TO clave
3023 GET 1,x
3025 clave=CVI(clave$);PRINT USING
"##";clave;
3026 PRINT TAB(8)denominacion$;TAB(
29)proveedor$;
3028 cantidad=CVI(cantidad$);PRINT
TAB(53) USING "##";cantidad;;exi
min=CVI(eximin$);PRINT TAB (61) US
ING "##";eximin;
3029 precuni=CVS(precuni$);PRINT TA
B(69) USING "##";precuni;;prec
to=CVS(preco$);PRINT TAB (79)USING
"##";preco
3040 NEXT Y
3050 PRINT FNlocate$(50,27,"pulse u
na tecla para continuar")
3060 opcion$=INKEY$;IF opcion$="" T
HEN 3060
3070 IF clave=0 THEN 3380
3080 PRINT clr$;b=b+100;IF clave<10
0 THEN a=clave+100 :GOTO 3380
3090 CLOSE;LPRINT CHR$(18);LPRINT C
HR$(12);GOTO 1150
3100 FOR x=1 TO clave
3110 GET 1,x
3120 clave=CVI(clave$);LPRINT TAB(5
) USING "##";clave;
3130 LPRINT TAB(20)denominacion$;TA
B(50)proveedor$;
3140 cantidad=CVI(cantidad$);LPRINT
TAB(75)USING "##";cantidad;;exi
min=CVI(eximin$);LPRINT TAB(95) US
ING "##";eximin;
3150 precuni=CVS(precuni$);LPRINT T
AB(105) USING "##";precuni;;prec
to=CVS(preco$);LPRINT TAB(115) US
ING "##";preco
3160 OPEN"R",1,"base"
3170 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS prectos$
3180 GET 1,999;clave=CVI(clave$)
3190 PRINT "clave";TAB(10)"denomina
cion";TAB(35)"proveedor";TAB(S8)"ca
ntidad";TAB(70)"exi/min"
3200 PRINT STRING$(90,"-")
3210 FOR x=1 TO clave
3220 GET 1,x
3230 clave=CVI(clave$);cantidad=CVI
(cantidad$);eximin=CVI(eximin$)
3240 IF cantidad<eximin THEN PRINT
clave;TAB(10)denominacion$;TAB(35)p
roveedor$;;PRINT TAB(60) USING "##";
##";cantidad;;PRINT TAB(70) USING
"##";eximin
3250 NEXT
3260 PRINT FNlocate$(50,27,"pulse u
na tecla para continuar")
3270 opcion$=INKEY$;IF opcion$="" T
HEN 3270
3280 CLOSE;GOTO 1150
3290 REM *** listado material por i
mpresora ***
3300 PRINT clr$
3310 OPEN"R",1,"base"
3320 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS prectos$
3330 GET 1,999;clave=CVI(clave$);a=
100;b=1
3340 LPRINT CHR$(15)
3350 LPRINT TAB(10)"clave";TAB(25)"d
enominacion";TAB(55)"proveedor";TA
B (75)"cantidad";TAB(95)"exi/min";T
AB(106)"prec/uni";TAB (116)"prec/ta
tal"
3360 LPRINT STRING$(130,"-")
3370 IF clave<a THEN 3500
3380 FOR y=b TO a
3390 clave=clave-1
3400 GET 1,y
3410 clave=CVI(clave$);LPRINT TAB(5
) USING "##";clave;TAB(20)denomin
acion$;TAB(50)proveedor$;
3420 cantidad=CVI(cantidad$);LPRINT
TAB(75) USING "##";cantidad;;exi
min=CVI(eximin$);LPRINT TAB(95) U
SING "##";eximin;
3430 precuni=CVS(precuni$);LPRINT T
AB(105) USING "##";precuni;;prec
to=CVS(preco$);LPRINT TAB(115) US
ING "##";preco
3440 NEXT Y
3450 PRINT FNlocate$(50,27,"pulse u
na tecla para continuar")
3460 opcion$=INKEY$;IF opcion$="" T
HEN 3460
3470 IF clave=0 THEN 3380
3480 PRINT clr$;b=b+100;IF clave<10
0 THEN a=clave+100 :GOTO 3380
3490 CLOSE;LPRINT CHR$(18);LPRINT C
HR$(12);GOTO 1150
3500 FOR x=1 TO clave
3510 GET 1,x
3520 clave=CVI(clave$);LPRINT TAB(5
) USING "##";clave;
3530 LPRINT TAB(20)denominacion$;TA
B(50)proveedor$;
3540 cantidad=CVI(cantidad$);LPRINT
TAB(75)USING "##";cantidad;;exi
min=CVI(eximin$);LPRINT TAB(95) US
ING "##";eximin;
3550 precuni=CVS(precuni$);LPRINT T
AB(105) USING "##";precuni;;prec
to=CVS(preco$);LPRINT TAB(115) US
ING "##";preco

```

```

SING"ffff";precio
3560 NEXT
3570 PRINT FN1locate$(50,27,"pulsa u
na tecla para continuar")
3580 opcion$=INKEY$:IF opcion$="" T
HEN 3580
3590 CLOSE:LPRINT CHR$(18);LPRINT C
HR$(12);GOTO 1150
3600 REM *** listado minimos por im
presora ***
3610 PRINT clr$
3620 LPRINT "clave"; TAB(10)"denomi
nacion";TAB(35)"proveedor";TAB(56)""
cantidad";TAB(68)"exi/min"
3630 LPRINT STRING$(80,"-")
3640 OPEN"R",1,"base"
3650 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS precio$
3660 GET 1,999;clave=CVI(clave$)
3670 FOR x=1 TO clave
3680 GET 1,x
3690 clave=CVI(clave$);cantidad=CVI
(cantidad$);eximin=CVI(eximin$)
3700 IF cantidad<eximin THEN LPRINT
clave;TAB(8)denominacion$;TAB(30)P
"proveedor$";LPRINT TAB(55)"USING ""R
R
R";cantidad;LPRINT TAB(65) USING
"R
R";eximin
3710 NEXT
3720 LPRINT CHR$(12)
3730 PRINT FN1locate$(50,27,"pulsa u
na tecla para continuar")
3740 opcion$=INKEY$:IF opcion$="" T
HEN 3740
3750 CLOSE;GOTO 1150
3760 REM *** inventariado ***
3770 PRINT clr$
3780 OPEN"R",1,"base"
3790 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS precio$
3800 GET 1,999;clave=CVI(clave$);a=
22;b=1
3810 IF clave<a THEN 4000
3820 PRINT"clave";TAB(10)"denominac
ion";TAB(32)"unidades";TAB(42)"prec
/uni";TAB(52)"prec/total"
3830 PRINT STRING$(65,"-")
3840 FOR y=b TO a
3850 clave=clave-1
3860 GET 1,y
3870 clave=CVI(clave$);PRINT USING
"R
R"; clave;
3880 PRINT TAB(10)denominacion$;
3890 cantidad=CVI(cantidad$);PRINT
TAB(35) USING "R
R";cantidad;
3900 precuni=CVS(precuni$);PRINT TA
B(40) USING "R
R";precuni;:pre
cto=CVS(precio$);PRINT TAB(50)USIN
"R
R";precio
3910 inventario=inventario+precio
3920 NEXT y
3930 PRINT;PRINT TAB(32)"total";:PR
INT TAB(50) USING "R
R";invent
ario
3940 PRINT TAB(32)"-----";inventario=0
3950 PRINT FN1locate$(50,29,"pulsa u
na tecla para continuar")
3960 opcion$=INKEY$:IF opcion$="" T
HEN 3960
3970 IF clave=0 THEN PRINT inventar
io;CLOSE;GOTO 70
3980 PRINT clr$;b=b+22:IF clave<22
THEN a=clave+22;GOTO 3820
3990 CLOSE;GOTO 70
4000 PRINT"clave";TAB(10)"denominac
ion";TAB(32)"unidades";TAB(42)"prec
/uni";TAB(52)"prec/total"
4005 PRINT STRING$(65,"-")
4010 FOR x=1 TO clave
4020 GET 1,x
4030 clave=CVI(clave$);PRINT USING
"R
R";clave;
4050 PRINT TAB(10)denominacion$;
4060 cantidad=CVI(cantidad$);PRINT

```

```

TAB(35) USING "R
R";cantidad;
4070 precuni=CVS(precuni$);PRINT TA
B(40) USING "R
R";precuni;:pre
cto=CVS(precio$);PRINT TAB(50)USIN
"R
R";precio
4080 inventario=inventario+precio
4090 NEXT
4100 PRINT;PRINT TAB(32)"total";:PR
INT TAB(50) USING "R
R";invent
ario
4110 PRINT TAB(32)"-----";inventario=0
4120 PRINT FN1locate$(40,29,"pulsa (i) para listar o una tecla para con
tinuar")
4130 opcion$=INKEY$:IF opcion$="" T
HEN 4130
4135 IF opcion$="i" OR opcion$="I"
THEN CLOSE;GOTO 4150
4140 CLOSE;GOTO 70
4150 REM *** listado inventario ***
4160 PRINT clr$
4170 OPEN"R",1,"base"
4180 FIELD 1,2 AS clave$,20 AS deno
minacion$,20 AS proveedor$, 2 AS ca
ntidad$,2 AS eximin$,4 AS precuni$,
4 AS precio$
4190 GET 1,999;clave=CVI(clave$);a=
60:b=1
4200 IF clave<a THEN 5000
4210 LPRINT"clave";TAB(10)"denomina
cion";TAB(32)"unidades";TAB(42)"pre
c/uni";TAB(52)"prec/total"
4220 LPRINT STRING$(65,"-")
4230 FOR y=b TO a
4240 clave=clave-1
4250 GET 1,y
4260 clave=CVI(clave$);PRINT USING
"R
R"; clave;
4270 LPRINT TAB(10)denominacion$;
4280 cantidad=CVI(cantidad$);LPRINT
TAB(35) USING "R
R";cantidad;
4290 precuni=CVS(precuni$);LPRINT TA
B(40) USING "R
R";precuni;:pre
cto=CVS(precio$);LPRINT TAB(50)USIN
"R
R";precio
4300 inventario=inventario+precio
4310 NEXT y
4320 LPRINT;LPRINT TAB(32)"total";:
LPRINT TAB(50) USING "R
R";inv
entario
4330 LPRINT TAB(32)"-----";inventario=0
4340 PRINT FN1locate$(50,29,"pulsa u
na tecla para continuar")
4350 opcion$=INKEY$:IF opcion$="" T
HEN 4350
4360 IF clave=0 THEN PRINT inventar
io;CLOSE;GOTO 70
4370 b=b+60;IF clave<60 THEN a=clav
e+60 ;GOTO 4000
4380 CLOSE;LPRINT CHR$(12);GOTO 70
5000 LPRINT"clave";TAB(10)"denomina
cion";TAB(32)"unidades";TAB(42)"pre
c/uni";TAB(52)"prec/total"
5005 LPRINT STRING$(65,"-")
5010 FOR x=1 TO clave
5030 GET 1,x
5040 clave=CVI(clave$);LPRINT USING
"R
R";clave;
5050 LPRINT TAB(10)denominacion$;
5060 cantidad=CVI(cantidad$);LPRINT
TAB(35) USING "R
R";cantidad;
5070 precuni=CVS(precuni$);LPRINT TA
B(40) USING "R
R";precuni;:pre
cto=CVS(precio$);LPRINT TAB(50)USIN
"R
R";precio
5080 inventario=inventario+precio
5090 NEXT
5100 LPRINT;LPRINT TAB(32)"total";:
LPRINT TAB(50) USING "R
R";inv
entario
5110 LPRINT TAB(32)"-----";inventario=0
5120 PRINT FN1locate$(50,29,"pulsa o
una tecla para continuar")
5130 opcion$=INKEY$:IF opcion$="" T
HEN 5130
5140 CLOSE;LPRINT CHR$(12);GOTO 70

```

GANA 100.000 PESETAS CON MICROHOBBY AMSTRAD SEMANAL

Porque pretendemos que **AMSTRAD SEMANAL** sea también vuestra revista, hemos abierto una sección en la que se publicarán los mejores programas originales recibidos en nuestra redacción. Vosotros seréis las encargadas de realizar estas páginas, en las que podréis oporturar ideas y programas interesantes para otros lectores.

Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.

- Cada programa debe ir acompañado de un texto explicativa en el cual se incluyan:

- Descripción general del programa.
- Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellos.
- Instrucciones de manejo.

- Todas estas datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.

- No se admitirán programas que contengan caracteres de control, debido a que no son correctamente interpretados por las impresoras.

- En una sola cinta puede introducirse más de un programa.

- Una vez publicado, **AMSTRAD SEMANAL** abonará al autor del programa de **15.000 a 100.000 pesetas**, en concepto de derechos de autor.

- Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.

- **AMSTRAD SEMANAL** se reserva el derecho de publicación o no del programa.

- Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANAL**.

- Los programas sospechosos de plagio serán eliminados inmediatamente.

¡ENVIANOS TU PROGRAMA!

Adjuntando los siguientes datos:

Nombre y apellidos,
dirección y teléfono.

Indicando claramente en el sobre:

AMSTRAD SEMANAL

a HOBBY PRESS, S. A. La Granja, 39
Pal. Ind. Alcabendas (Madrid)

ULTIMA HORA

MAS ACERCA DEL PCW8512

D

espués de dar a nuestros lectores la exclusiva de la aparición del nuevo Amstrad, el **PCW8512** tenemos un poco más de información que ofreceros.

La nueva máquina, recordemos, tiene 512 kbytes de RAM, e incorpora una segunda unidad de disco de un Megabyte (*todavía no sabemos si formateado el disco o no*).

Del medio mega de RAM, 364 K se dedican precisamente a RAMDISK, esto es, un disco virtual donde pueden almacenarse y manipularse ficheros como en los discos «normales», pero con una abismal diferencia de velocidad.

El precio del **PCW8512**, según una nota oficial de prensa de Indescomp, será de 174.900 pesetas sin IVA.

Los usuarios del **PCW8256** pueden «trasladarse» a la nueva máquina mediante un kit que, siempre según Indescomp, ya está disponible al precio **47.500** ptas. (*IVA incluido*); se compone de los chips de memoria y de la segunda unidad de disco.

Respecto al conflictivo asunto de qué va a pasar con el **PCW8256**, el director comercial de Indescomp, Miguel Angel Esteban, manifestó textualmente:



«El 8256 seguirá siendo la estrella de nuestros ordenadores profesionales, pero hay un sector de usuarios que busca un equipo con mayor capacidad y el PCW8512 es la respuesta.»

E

l **PCW8512** se vende al público con el mismo software de base que el 8256, es decir: Mallard Basic, sistema operativo CP/M+ y DR Logo.

La unidad de disco posee un Megabyte sin formatear, y proporciona 720 Kbytes una vez formateada, en doble densidad.

Al parecer, la memoria extra y las dos unidades de disco pueden conseguir que el Lacoscript examine ortográficamente 25 páginas por minuto.

Igualmente, será posible mantener en la memoria varios programas diferentes a la vez, pero desconocemos si serán capaces o no de ejecutarse concurrentemente.

En Inglaterra, el nuevo 8512 estará en las tiendas por estas fechas, y su precio será más o menos de 575 #, unas **115.000** pesetas (*IVA incluido*).



VOLVIENDO AL ABACO

El aspecto educativo de la informática cada vez tiene mayor importancia; nadie duda, hoy en día, que los ordenadores son una poderosa herramienta de aprendizaje.

En Amstradideas también lo sabemos, y por eso presentamos un programa especialmente dedicado a los más pequeños para que sin esfuerzo, puedan aprender los secretos de los números.

Lourdes Biec Arbués

E

l programa **ABACO**

es muy sencillo y está orientado para el Ciclo Inicial. Su objetivo es representar números de hasta 3 cifras mediante el abaco; identificando cada cifra con el número de bolas que indica.

```

1 MODE 1:INK 1,6,22:INK 2,24
2 PEN 1:LOCATE 10,1:PRINT "*** A B
A C D ***":PEN 2:PRINT:PRINT:PRINT
"ESTE PROGRAMA ESTÁ DESTINADO PARA
CHAVAS-LITOS DE PARVULOS Y CICLO INI
CIAL."
3 PRINT:PRINT:PRINT "CONSISTE EN RE
PRESENTAR LAS UNIDADES, --DECENAS Y
CENTENAS MEDIANTE BOLAS EN EL ABAC
O."
4 PRINT:PRINT:PRINT "EL PROGRAMA TE
PREGUNTA LAS UNIDADES CON UNA 'U',
LAS DECENAS CON UNA 'D' Y LAS
CENTENAS CON UNA 'C'."
5 PRINT:PRINT:PRINT "EL SIGUIENTE P
ROGRAMA FUE REALIZADO POR:"PEN 1:L
OCATE 10,20:PRINT "LOURDES BIEC (HU
ESCA)"
6 PEN 2:LOCATE 1,24:PRINT "Pulsa un
a tecla para continuar"
7 CALL &BB06
8 MODE 0:INK 0,13:BORDER 13:INK 1,6
:INK 2,0:INK 3,1B:INK 4,17:PRINT CH
R$(22)+CHR$(1)
9 PEN 1:LOCATE 4,1:PRINT "A B A
C D":LOCATE 4,2:PRINT "-----"
10 PLOT 80,160:DRAWR 448,0,2:PLOT 1
44,160:DRAWR 0,176:PLOT 304,160:DRA
WR 0,176:PLOT 464,160:DRAWR 0,176
11 PEN 1:LOCATE 15,17:INPUT "U":u:LOC
ATE 10,17:INPUT "D":d:LOCATE 5,17
:INPUT "C":c
12 PEN 3:LOCATE 2,21:PRINT "NUMERO:
";c;d;u
13 z=u+w+l:w=15-d+c+f=15-c+1
14 FOR y=15 TO z STEP -1
15 LOCATE 15,y:PRINT CHR$(231)
16 SOUND 1,500,50:FOR n=1 TO 1000:N
EXT n,y
17 FOR y=15 TO w STEP -1
18 LOCATE 10,y:PRINT CHR$(231)
19 SOUND 1,500,50:FOR n=1 TO 1000:N
EXT n,y
20 FOR y=15 TO f STEP -1
21 LOCATE 5,y:PRINT CHR$(231)
22 SOUND 1,500,50:FOR n=1 TO 1000:N
EXT n,y
23 PEN 4:LOCATE 1,25:INPUT "OTRO? "
S/N)":a
24 IF a$="s" OR a$="S" THEN B
25 IF a$="n" OR a$="N" THEN MODE 1:
INK 0,1:INK 1,24:Pen 1:BORDER 1:END

```

MERCENARIO

© ALLIGATA SOFTWARE

para **AMSTRAD**

Vive una gran aventura.

Cada pantalla es un nuevo desafío al peligro, una misión suicida, una dura batalla. Te encontrarás solo frente a combates de asalto, gitanos, morteros, cañones de mortero, etc., que emplearán todo su fuerza y material bélico para frenar tu avance al demolido. Hace falta mucha sangre y mucha agilidad mental para librarse del ataque más letal o las bombas bajas.

visto

Cada partida es un mundo a descubrir.

cassette: 2.000,- Ptas. (+I.V.A.)

disco: 2.900,- Ptas. (+I.V.A.)

PRODUCE Y
DISTRIBUYE

micro 64

P.º CASTELLANA, 179-1.º - 28046 MADRID Tel. 442 54 33/44



Mercado común

Con el objeto de fomentar las relaciones entre los usuarios de AMSTRAD, MERCADO COMUN te ofrece sus páginas para publicar los pequeños anuncios que relacionados con el ordenador y su mundo se ajusten al formato indicado a continuación.

En MERCADO COMUN tienen cabido, anuncios de ventas, compras, clubs de usuarios de AMSTRAD, programadores, y en general cualquier clase de anuncio que pueda servir de utilidad a nuestros lectores.

Envianos tu anuncio mecanografiado a: HOBBY PRESS, S.A.
AMSTRAD SEMANAL.
Aportado de correos 54.062
28080 MADRID
¡ABSTENERSE PIRATAS!

Se intercambian juegos para el Amstrad CPC-464 con usuarios de Barcelona a provincia. Llamar de 15,30 a 17 h o de 21 a 22 h al Tel. 212 42 84. Preguntar por Iván.

Cambio ordenador Amstrad CPC-464 monitor color en perfecto estado, más 30 programas comerciales, por Amstrad CPC-664 monitor color pagando diferencia justa. Llamar al Tel. (93) 212 42 84 de Barcelona. Preguntar por Iván. (Llamar de 21,30 a 22,00 h.).

Desearía contactar con usuarios del Amstrad CPC-6128, en Cádiz y su provincia. Interesados escribir a: Silvia Rodríguez Fernández. Avda. Ramón de Carranza, 22 - P. 6.º D. 11006 Cádiz. Tel. 28 72 80.

Vendo impresora Centronics 730-2, matricial, tracción y fricción, 80 —40 a doble anchura—, a 132 columnas —66 a doble anchura—, 80 caracteres por segundo. Regalo además 500 hojas de papel continuo y 3 cintas de recambio (originales). Con cable y conector para el Amstrad: 33.000 ptas. Con conector estándar: 30.000 ptas. Interesados llamar al Tel. (91) 742 28 37, preferiblemente de 10 a 14 h. Claudia Pérez-Olea. C/ Telémaco, 26. 28027 Madrid.

Desearía contactar con usuarios/as de ordenadores en general y de cualquier zona del mundo. Interesados/as escribid a: Ramón J. Puchades. C/ Fco. Climent, 6 - 1.º. 46007 Valencia.

Cambio juegos comerciales para Amstrad (en cinta) preferentemente con gente de Barcelona. Entre ellos están Exploding Fist, Combat Linx, Decathlon, copiones, etc. Alberto Adeva Antón. C/ Eduardo Todá, 36 int. 5.º-1.º 08031 Barcelona. Tel. 427 08 32.

Vendo Amstrad CPC-464, color. En perfecto estado, con manuales y los programas siguientes:

- Ensamblador y desensamblador Devpac de Amsoft.
- Battle for midway.
- Hunter killer.
- Roland in the caves.
- Combat lynx.

Precio actual de todo, 111.000 ptas. apróx. y lo vendo por 80.000 ptas. Llamar a partir de las 18,00 h. y preguntar por María.

Desearía contactar con usuarios del Amstrad CPC-464. José María Gómez Vega. P.O.BOX-381. Pto. Santa María (Cádiz).

Vendo curso Autodidáctico de Basic Tomos I y II, con dos cassettes para prácticas cada uno. Los dos por 3.500 ptas. Juan Puig Arnau. C/ P.º Ramón Vall, 44 - 3.º-2.º. Navas (Barcelona).

Desearía adquirir Amstrad de segunda mano. Interesados dirigirse a D. G. Learza. C/ Don Vela, 24 - 3.º Izqda. 01009 Vitoria (Alava).

Intercambio toda clase de programas. Interesados dirigirse a Luis García Cantabrana. C/ Argentina, 14 - 3.º A. 01009 Vitoria (Alava).

MICRO DEALER

CENTRO DE DISTRIBUCIÓN ESPECIALIZADA EN AMSTRAD

Comandante Zorita, 13
Tel. 233 07 35 - 07 81
MADRID 28020

TODO EN:

Joysticks	Impresoras
Cables	Interfaces
Monitores	Ampliaciones de memoria
Diskettes 3"	Teclados profesionales
Cassettes turbo	Cintas vírgenes
Unidades de disco	Diskette 5½ DCDD

MAYORISTAS DE:

Amstrad	Sinclair
Commodore	Spectravideo
Star	Newprint
Seikosa	Dk'tronics

NOVEDAD

Reserve su
Amstrad PCW8256
PVP 129.900

PRECIOS DE OFERTA EN IMPRESORAS:

20% Dto. sobre PVP
¡¡INFORMATE!!

**¡¡BUSCAMOS DISTRIBUIDORES
EN PROVINCIAS!!**

**PIDA LISTA DE PRECIOS
INFORMATE EN MICRO DEALER:
Tel. (91) 233 07 81
(915) 23 30 35**

MICRO-1

el IVA lo paga
MICRO-1

C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 275 96 16/274 53 80
(Metro O'Donell o Goya)

SOFTWARE: por cada programa GRATIS ¡¡1 BOLIGRAFO CON RELOJ DE CUARZO!!

HYPER SPORTS	2.300 ptas.
TORNADO LOW LEVEL	1.950 ptas.
EXPLODING FISTT	2.300 ptas.
JUMP JET	2.495 ptas.
ZORRO	2.600 ptas.
SABREWULF	1.650 ptas.
GHOSTBUSTERS	1.950 ptas.
GYROSCOPE	2.300 ptas.
HYGHWAY ENCOUNTER	1.750 ptas.
HIGHWAY ENCOUNTER DISCO	3.300 ptas.

DYNAMITE DAN	2.100 ptas.
RAID OVER MOSCOW	2.300 ptas.
THEY SOLD A MILLION	2.500 ptas.
FIGHTER PILOT	1.975 ptas.
MASTER OF T. LAMP	1.950 ptas.
NIGHTSHADE	1.950 ptas.
HACKER	1.950 ptas.
SUPER TEST	2.300 ptas.
MAPGAME	2.700 ptas.
TONADO LOW LEVEL DISCO	3.300 ptas.

JOYSTICK QUICK SHOTT II... 2.295 ptas.
JOYSTICK QUICK SHOT V ... 2.595 ptas.

PC-COMPATIBLE IBM 256 K
MONITOR FOSFORO VERDE
2 BOCAS DISKETTE 360 K
SOLO ¡¡243.900!!

TAPA METACRILATO PARA
TECLADO ¡¡1.900 ptas.!!

UNIDAD DISKETTE 5.25"
¡¡45.900 ptas.!!
(incluido controlador)

LAPIZ OPTICO
¡¡4.900 ptas.!!

IMPRESORA MARGARITA
¡¡49.900 ptas.!!

CASSETTE ESPECIAL
ORDENADOR 5.295 ptas.

PRECIOS SUPER-EXCEPCIONALES PARA
AMSTRAD CPC-472 Y CPC-6128
¡¡LLAMANOS, TE ASOMBRARAS!!

IMPRESORAS
¡¡20% DTO. SOBRE P.V.P.!!

SINTETIZADOR DE VOZ
Y AMPLIFICADOR:
7.900 ptas.

MODULADOR TV
8.400 ptas.

INTERFACE DISCO
5 1.4" 5.300 ptas.

CINTA C-15 ESPECIAL
ORDENADOR 85 ptas.
DISKETTE 3" 990 ptas.

Libros:
Curso autodidáctico Basic I 2.525 ptas.
Curso autodidáctico Basic II 2.525 ptas.
Programando con Amstrad 2.195 ptas.
Juegos sensacionales Amstrad 1.950 ptas.
Hacia la Inteligencia Artificial 1.295 ptas.
Música y sonidos con Amstrad 995 ptas.

UNIDAD DE DISCO 3" CON
CONTROLADOR: 49.900 ptas.

RAMBO

FIRST BLOOD PART II



OTRA
EXCLUSIVA

ocean

EIDOS Software

SANTA ENGRACIA, 17. 28010 MADRID. Tel.: 447 34 10